Computational Studies in Multi-Criteria Scheduling and Optimization

Megan Wydick Martin

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Business Information Technology

Cliff T. Ragsdale, Chair
Jason K. Deane
Lance A. Matheson
Terry R. Rakes
Roberta S. Russell

July 2017
Blacksburg, Virginia

Key words: Multi-Criteria Scheduling, Workload Balancing, Task Assignment, Tour Scheduling,
Test Scheduling, Optimization

Computational Studies in Multi-Criteria Scheduling and Optimization

Megan Wydick Martin

ABSTRACT

Multi-criteria scheduling provides the opportunity to create mathematical optimization models that are applicable to a diverse set of problem domains in the business world. This research addresses two different employee scheduling applications using multi-criteria objectives that present decision makers with trade-offs between global optimality and the level of disruption to current operating resources. Additionally, it investigates a scheduling problem from the product testing domain and proposes a heuristic solution technique for the problem that is shown to produce very high-quality solutions in short amounts of time.

Chapter 2 addresses a grant administration workload-to-staff assignment problem that occurs in the Office of Research and Sponsored Programs at land-grant universities. We identify the optimal workload assignment plan which differs considerably due to multiple reassignments from the current state. To achieve the optimal workload reassignment plan we demonstrate a technique to identify the *n* best reassignments from the current state that provides the greatest progress toward the utopian solution. Solving this problem over several values of *n* and plotting the results allows the decision maker to visualize the reassignments and the progress achieved toward the utopian balanced workload solution.

Chapter 3 identifies a weekly schedule that seeks the most cost-effective set of coach-to-program assignments in a gymnastics facility. We identify the optimal assignment plan using an integer linear programming model. The optimal assignment plan differs greatly from the status quo; therefore, we utilize a similar approach from Chapter 2 and use a multiple objective optimization technique to identify the *n* best staff reassignments. Again, the decision maker can

visualize the trade-off between the number of reassignments and the resulting progress toward the utopian staffing cost solution and make an informed decision about the best number of reassignments.

Chapter 4 focuses on product test scheduling in the presence of in-process and at-completion inspection constraints. Such testing arises in the context of the manufacture of products that must perform reliably in extreme environmental conditions. Each product receives a certification at the successful completion of a predetermined series of tests. Operational efficiency is enhanced by determining the optimal order and start times of tests so as to minimize the make span while ensuring that technicians are available when needed to complete in-process and at-completion inspections We first formulate a mixed-integer programming model (MILP) to identify the optimal solution to this problem using IBM ILOG CPLEX Interactive Optimizer 12.7. We also present a genetic algorithm (GA) solution that is implemented and solved in Microsoft Excel. Computational results are presented demonstrating the relative merits of the MILP and GA solution approaches across a number of scenarios.

Computational Studies in Multi-Criteria Scheduling and Optimization

Megan Wydick Martin

GENERAL ABSTRACT

Multi-criteria scheduling provides the opportunity to create mathematical optimization models that are applicable to a diverse set of problem domains in the business world. This research addresses two different employee scheduling applications using multi-criteria objectives that present decision makers with trade-offs between global optimality and the level of disruption to current operating resources. Additionally, it investigates a scheduling problem from the product testing domain and proposes a heuristic solution technique for the problem that is shown to produce very high-quality solutions in short amounts of time.

Chapter 2 addresses a grant administration workload-to-staff assignment problem that occurs in the Office of Research and Sponsored Programs at land-grant universities. Solving this problem and plotting the results allows the decision maker to visualize the number of reassignments and the progress achieved toward the utopian balanced workload solution. Chapter 3 identifies a weekly schedule that seeks the most cost-effective set of coach-to-program assignments in a gymnastics facility. Again, the decision maker can visualize the trade-off between the number of reassignments and the resulting progress toward the utopian staffing cost solution and make an informed decision about the best number of reassignments.

Chapter 4 focuses on product test scheduling in the presence of in-process and at-completion inspection constraints. Such testing arises in the context of the manufacture of products that must perform reliably in extreme environmental conditions. Each product receives a certification at the successful completion of a predetermined series of tests. Computational

results are presented demonstrating the relative merits of the mixed integer linear programming model and the genetic algorithm solution approaches across a number of scenarios.

Dedication

"Alone we can do so little; together we can do so much." -Helen Keller

I dedicate my work to my family and friends. A special feeling of gratitude to my mother, Beth Tessier, whose unlimited generosity provided me with the time and sanity to complete my dissertation. To my children, Bryce and Brooke Anne Martin, who drive my internal cheerleader to accept nothing less than excellence in a humble effort to set a good example.

And to my loving husband, Keith Martin, for his unending support and encouragement. I am so grateful for the many hours of listening, brainstorming, and proofreading a field of research that was not always your favorite. You never left my side and are truly my better half.

Acknowledgements

"Optimism is the faith that leads to achievement. Nothing can be done without hope and confidence." -Helen Keller

I am thankful to the members of my committee for their patience and support in overcoming numerous obstacles I have been facing through my research.

Extreme gratitude goes to my advisor, Cliff Ragsdale, for his honesty, encouragement and endless patience. It was a pleasure to have the opportunity to work with such an educator as a mentor. For empowering me to think about concepts and use my mind in ways that were unimaginable. Thank you for your support, because without you, I would not be writing this today.

Table of Contents

List of Tables

List of Figures

Chapter 1 : Introduction and Literature Review

1.0 Introduction

Multi-criteria scheduling provides the opportunity to create mathematical optimization models that are relevant and applicable to a variety of problems in the business world. Business decisions are usually not driven by one objective but rather multiple objectives that can conflict with one another. This research addresses two applications of employee scheduling using multi-criteria objectives to identify solutions that improve business operations. Whether the decision maker needs to balance employee workload, minimize the number of employee reassignments to tasks, reduce staffing cost or improve production time, the struggle to meet multiple objectives is constant. Additionally, we investigate a scheduling problem from the product testing domain and proposes a heuristic solution technique for the problem that is shown to produce very high-quality solutions in short amounts of time.

Chapter 2 addresses a recurring staffing issue that occurs in the Office of Research and Sponsored Programs at a large, public, land-grant university in the pre-award process, which includes the application and review portion of sponsored research. We first identify the optimal (utopian) workload assignment plan using a mixed-integer linear programming problem. Unfortunately, the optimal assignment of ORSP staff members to departments in this problem differs considerably from the status quo, requiring multiple reassignments from the current state to reach optimality. The number of reassignments raises concerns related to loss of administrator-department relationship, loss of department-related knowledge, and increase in inconvenience.

To achieve the best workload reassignment with the fewest changes from the current status quo, while still placing a greater emphasis on the effective use of limited resources, we propose and illustrate a multiple objective optimization technique to identify the *n* best departmental reassignments from the current state that provide the greatest progress toward the utopian solution. Solving this problem over several values of *n* and plotting the results allows the decision maker to visualize the trade-off between the number of reassignments and the resulting progress achieved toward the utopian balanced workload solution.

Chapter 3 considers the unique challenges of personnel task assignment and tour scheduling using a cross-trained workforce within a gymnastics facility. Scheduling a sufficient number of employees for each shift for each day, along with the time-on-task assignment over a 7-day operation while ensuring all employees scheduled are qualified to perform that task, creates a distinct and innovative model in personnel scheduling. Typically, gymnastic facility business owners require 24+ hours to complete the formulation of an 8-week personnel schedule, which is not easily adjusted due to the complexity of hard copy calculations, personnel job qualifications, coach-to-program ratios and coach-to-gymnast established relationships (Kennedy, 2017; Tessier, 2017). We propose a model for determining a weekly personnel schedule that seeks the most cost-effective set of coach-to-program assignments in a gymnastics facility using an integer linear programming model.

The optimal (utopian) coach-to-program assignment plan differs considerably from the status quo and requires a large number of coach-to-program reassignments to reach optimality. We employ a similar approach from Chapter 2 and use a multiple objective optimization technique to identify the *n* best coach-to-program reassignments from the current state. Making these adjustments with minimal coach reassignments helps the children feel at ease with familiar coaches and maintains a level of staffing cost that is ideal for the decision maker. Again, the decision maker can visualize the trade-off between the number of reassignments and the resulting progress achieved toward the utopian staffing cost and make an informed decision about the best number of reassignments.

Chapter 4 focuses on the scheduling of in-process and at-completion testing of industrial equipment used in extreme environmental conditions. As the recent Deepwater Horizon oil spill illustrates, failure of industrial equipment in sensitive environmental settings can have devastating impacts on the natural world as well as severe economic implications for the associated business entities. Thus, the scheduling of extreme environmental conditions testing on products for market certification presents both a complex and critically important problem. Products might need to be exposed to low and high temperatures as well as variations of pressure and humidity in isolated chambers as part of the certification process. Thus, the certification process might a series of in-process and at-completion inspections by personnel to validate the functionality and durability of the product before it is delivered and deployed. Each test requires a certain amount of time under harsh conditions with inspections at the end of each test along with some tests also requiring inspections during of the testing cycle.

We propose a methodology for identifying the optimal order and start times of tests that minimizes the make span while ensuring the necessary employees are present to complete the in-process and at-completion inspections, resulting in increased operational efficiency. We first present a mixed-integer linear programming model for this problem that is easily solved to global optimality using IBM ILOG CPLEX Interactive Optimizer 12.7. Given the cost of specialized integer programming software, we next identify a genetic algorithm-based heuristic that may be used to solve the problem with Microsoft Excel 2016. We provide computational results comparing the results of the mixed-integer linear programming model and the genetic algorithm.

2.0 Literature Review

The nature of the work presented in this research leads to an evaluation of literature in multi-criteria scheduling related to personnel scheduling as well as product test scheduling. A brief overview of these research fields is provided to support the methodology and techniques used to develop the integer linear programming models and the genetic algorithm solution methodology.

Multi-criteria decision making can be categorized in two areas: multi-objective decision making and multi-attribute decision making. Multi-objective decision making is known as the design problem where there is a large, implicit set of feasible alternatives. Multi-attribute decision making is known as the choice problem where there is a small, explicit set of feasible alternatives (Hwang, Paidy, Yoon, & Masud, 1980; Steuer, 1986; Yoon & Hwang, 1995). We will concentrate on the multi-objective decision making categorization for the literature review as it relates to our research.

A basic taxonomy structure was developed by Hwang, Paidy, Yoon and Masud and supported by Evans and Marler and Arora (Hwang et al., 1980; Evans, 1984; Marler & Arora, 2004). The taxonomy is organized into four categories based on the preference information provided by the decision maker. In other words, the categorization is dependent on the timing and interaction of the decision maker with the solution methodology. The categories include: a priori, progressive, a posteriori, and no articulation.

A priori is when the researcher receives the preference information from the decision maker before the model is solved or optimized. In other words, the researcher has an idea of the decision maker’s preference levels or trade-offs. Progressive an interactive exchange concerning the decision maker's preferences during the model optimization process. Model adjustments at each iteration are made based on the feedback and interaction of the decision maker during the solution process. This approach is less common due to the time-consuming, interactive requirement on the decision maker.

A posteriori is when the methodology provides a range of solutions to the decision maker after analyzing a model and the decision maker selects the preferred solution. The decision maker evaluates a trade-off between objectives based upon some non-quantifiable criteria. The no articulation categorization occurs when there is minimal interaction with the decision maker and the solution methodology at any point in the process. The decision maker enjoys little disturbance but has little involvement and the solution methodology might need to make assumptions without input from the decision maker. In our multi-criteria scheduling and optimization research we utilize decision making models that can be categorized as a posteriori integer linear programming.

2.1 Personnel Task Assignment and Tour Scheduling

This research addresses personnel scheduling problems in the pre-award process in a grant administration department at a large, public, land-grant university and in coach-to-program task assignment and tour scheduling issues using a cross-trained workforce within a gymnastics facility. These can quickly become complex task assignment problems based on the size of the problem and integrality requirements.

Two early articles used probability theory and linear programming to address the task assignment problem (Dantzig, 1954; Edie, 1954). Later, Kuhn proposed the well-known Hungarian polynomial algorithm to solve the assignment problem (Kuhn, 1955; 1956). The simple assignment problem assessed a set of *n* personnel, *n* job tasks, and individual task qualifications to identify the best set of personnel-to-task assignments that maximizes the number of tasks completed while allowing only one task to one person (Votaw Jr. & Orden, 1951; Kuhn, 1955). The personnel-task qualifications were identified in a binary fashion with a 1 if the person was qualified to complete the task and 0 otherwise. While new algorithms have been developed that perform better in specific cases, the Hungarian method is still in common use.

In comparison, the general assignment problem enhanced the simple assignment problem by creating a rating matrix rather than the personnel-task qualifications and strived to maximize the sum of the ratings (Votaw Jr. & Orden, 1951; Kuhn, 1955). The rating matrix utilized positive integers to rate the personnel based on the task rather than a 0-1 classification. In 1951, Votaw and Orden presented several methods to address the general assignment problem, primarily concentrating on the simplex method, with satisfactory solutions but computational capacity constraints (Votaw Jr. & Orden, 1951).

Later, the general assignment problem evolved into the generalized assignment problem (GAP) by introducing a cost matrix, allowing multiple tasks to one person, and providing the best set of personnel-to-task assignments based on the minimum total cost (Ross & Soland, 1975; Fisher, Jaikumar, & Van Wassenhove, 1986). Let the binary variable $x_{ij} = 1$ if person *i* is assigned to task *j*; and 0 otherwise. The cost incurred if person *i* is assigned to task *j* is denoted by c_{ij}. The resources required by agent *i* to complete task *j* is denoted r_{ij} and the total amount of resource available for person *i* is identified by b_i.The standard formulation of a GAP is the following (Ross & Soland, 1975; Oncan, 2007). Note, the task assignment problem is formulated identical to the GAP with the addition of communication cost in the objective function (Li, Huang, & Fang, 2013).

$$\text{MIN} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

Subject To:

$$\sum_{j=1}^{n} r_{ij} x_{ij} \leq b_i, i = 1, \dots, m \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1,\ j = 1, \dots, n \tag{3a}$$

$$x_{ij} \in \{0,1\}, i = 1, \dots, n \ and\ j = 1, \dots, m \tag{4}$$

Given the nature of the GAP being an NP-hard combinatorial optimization problem, Dakin suggested a tree-search algorithm in 1965 motivated by Land and Doig to address such a problem (Land & Doig, 1960; Dakin, 1965; Fisher et al., 1986). Later, Ross and Soland expand on the tree-search method and present the branch and bound method to improve computational efficiency for the GAP (Ross & Soland, 1975).

In 1992, Cattrysse and Van Wassenhove provided a review of algorithms that address the GAP utilizing different relaxation techniques developed in the 1970’s and 1980’s (Cattrysse & Van Wassenhove, 1992). In 2007, Oncan provided a survey of the GAP along with current solution procedures including enhanced relaxation techniques in addition to heuristic and metaheuristic procedures (Oncan, 2007). Once again, relaxation techniques and heuristic procedures are used in an attempt to improve computational time, although, integer optimal solutions may be difficult to identify.

In addition to various methods to solve the GAP, variations or extensions to the mathematical structure of the GAP have been presented. In 1998, Park et. al presented the generalized multi-assignment problem by altering equation (3a) (Park, Lim, & Lee, 1998). This formulation allows multiple agents to be assigned to a single task. In the event $a_j = 1$ for all j, the formulation becomes the GAP.

$$\sum_{i=1}^{m} x_{ij} \geq a_j \quad j = 1, \dots, n \tag{3b}$$

In 2013, Li et. al. suggested a logarithmic approach to the GAP to reduce the number of binary variables and inequality constraints. This variation is used to improve computational time in large-sized GAPs. In an effort to maintain simplicity, we concentrate on the mathematical structure of the GAP demonstrated in Equations (1), (2), (3b) and (4).

The features of the gymnastics personnel scheduling model include shift scheduling (or hours of the day) and days off scheduling (or days of the week) (Baker, 1976; Bechtold, Brusco, & Showalter, 1991; Alfares, 2004; Brucker, Qu, & Burke, 2011; Van den Bergh, Beliën, De Bruecker, Demeulemeester, & De Boeck, 2013). In 1977, Henderson and Berry applied a branch and bound algorithm using an integer linear programming model to develop a personnel schedule for telephone operators for 15 minute shifts in a 24 hour timeframe (Henderson & Berry, 1977). The employees were assumed to be available for all shifts, ignoring variability of employee availability or employee job qualification.

In 1990, Thompson compared two linear programming models in regard to shift scheduling where the staffing pool is only available for particular shifts and where employees are qualified to perform all job tasks (Thompson, 1990). Thompson concentrated specifically on the management of services versus the management of production. We assume all employees are available for all shifts but qualified to perform one, some, or all job tasks.

A key contribution of the model presented in this research for the gymnastics scheduling problem is the utilization of cross-trained employees and the impact on staffing costs. We use a mixed, heterogeneous workforce where full time and part time employees are qualified to perform multiple job tasks within a facility but not necessarily all job tasks (Alfares, 2004).

Mahhotra and Ritzman utilized a simulation model to assist in the development of a personnel scheduling model concentrating exclusively on part time, cross-trained employees (Malhotra & Ritzman, 1994). They concluded that the benefit of part time, cross-trained employees is dependent on different operational factors; although, the overall impact of utilizing part time, cross-trained employees reduced staffing cost. We expand the scope of our model to part time and full-time cross-trained employees.

Brusco and Johns developed a personnel scheduling model using an integer linear program to minimize staffing costs and incorporating cross-trained employees within a single work shift (Brusco & Johns, 1998). They concluded that having a cross-trained workforce allows for better scheduling flexibility; although, they did not look at job qualification of each employee. We expand the problem to a 7-day planning period using multiple shifts in a 12-hour workday with employee availability parameters.

In 2002, Gomar, Haas and Morton evaluated the effect of using a cross-trained workforce in the construction industry where multiple projects with a variety of skills need to be completed (Gomar, Haas, & Morton, 2002). Their objective minimized employee turnover costs (hiring and firing) and found that cross-trained employees were always preferred. In constrast, the model presented here minimizes total staffing cost related to the payrate of the employee and the hours worked by that employee.

Later, Say and Karabat apply a two-stage optimization model to evaluate the trade-off between the output of the company and the skill improvement of the employee when using cross-trained employees (Sayın & Karabatı, 2007). Say and Karabat concentrate primarily on employee skill improvement. The present research utilizes the flexibility of a cross-trained staffing pool but concentrate on the staffing cost and the path to optimality of coach-to-program reassignments.

More recently, Rahimian, Akartunali and Levine implement a hybrid approach using integer programming and variable neighborhood search algorithms to address the nurse rostering problem (Rahimian, Akartunali, & Levine). The nurse rostering problem has similar features to the gymnastics scheduling problem including a restricted number of shifts scheduled in a day, employee availability constraints and a limited total number of workdays in the planning period. We incorporate different job tasks and allow the employee to change tasks multiple times within the scheduled work day. In addition, we will address our personnel scheduling problem using an integer programming model to allow for realistic modeling of the level of complexity in the structure of the problem.

In 2004, Naus presented the elastic generalized assignment problem which allows the agents to exceed or violate the resource capacity in equation (2) by using over time and under time. Alterations to equations (1) and (2) must be made to introduce the slack and surplus variables with the additional cost of over time or under time. This leads us to the grant administration personnel scheduling objective of balancing workload (or scheduled personnel-to-task assignment) to ensure employees are worked equally.

2.2 Load Balancing

The load balancing problem and its underlying problem of task assignment are well-known as it has long been a troubling issue. Load balancing is similar to line balancing in terms of distributing the work across the workstations equally; however, line balancing also emphasizes workstation precedence constraints (Casavant & Kuhl, 1988; Becker & Scholl, 2006; Gen, Cheng, & Lin, 2008). Line Balancing, also known as assembly line balancing, dates back to automobile manufacturing with Henry Ford in 1913 (Becker & Scholl, 2006; Gen et al., 2008). Specifically, tasks are assigned to workstations in a sequential order on an assembly line to meet one or more objectives.

Salveson formulated the first mathematical line balancing problem in a manufacturing context in 1954 (Ghosh & Gagnon, 1989; Battaia & Dolgui, 2013). Since then, more modern manufacturing systems appeared with disassembly, parallel lines, or workstations operated by computer-controlled robots (Becker & Scholl, 2006; Battaia & Dolgui, 2013). This has kept the line balancing problem of designing or re-designing the line of current interest to both researchers and practitioners.

In 2006, Harvey et. al. evaluated the load balancing problem which is an extension of the GAP (Harvey, Ladner, Lovász, & Tamir, 2006). Typically, the term load balancing is used in the context of computer processors, distributed systems, and project assignment (Chou & Abraham, 1982; Zhou, 1988; Liang, Li, Lim, & Guo, 2010; Penmatsa & Chronopoulos, 2011; Ali & Khan, 2012; Neelakantan, 2012). This research concentrates on the idea of reallocating workload from nodes exceeding capacity to idle or under-utilized nodes to meet or exceed specific performance criteria.

2.3 Optimality Assessment

The task assignment, tour scheduling and workload balancing algorithms have one point in common; they seek for an optimal solution. At times, knowing what is optimal is not sufficient and decision makers are interested in exploring suboptimal solutions due to other issues associated with application, implementation, and unarticulated objectives. Most assessments of suboptimal versus optimal solutions are focused on a cost or utility, not the actual number of reassignments.

In Murty's 1968 article, a partitioning technique is used to rank the *k* best assignments in order of increasing cost (Murty, 1968). Although the research is 45 years old, the technique is still used today. In 2008, small modifications of Murty's technique led to the development of an algorithm with a sub-optimal starting point and a search method for the best improvement rather than the lowest degradation (Pedersen, Relund Nielsen, & Andersen, 2008). However, that would give us the *k*-best solutions, but does not directly provide information on the number of reassignments required.

In 2004, Zulch, Rottinger, and Vollstedt acknowledged the issue of reassigning personnel to tasks, specifically in a manufacturing environment (Zülch, Rottinger, & Vollstedt, 2004). The goal is to reassign personnel to improve the overall utility which includes average lead time, system output, average workload, and labor costs; however, the individual cost of reassignment is not evaluated. A few years later, Gamberini, Grassi, and Rimini evaluate the cost of reassignment when rebalancing of a production line is needed due to changes in parameters

(Gamberini, Grassi, & Rimini, 2006). In their model, the rebalance of the line creates a solution that optimizes the line and reduces the cost of reassignment. This logic may reduce the number of reassignments realized, but the decision is not based on the number of reassignments desired. If the cost of reassignment is not known, the technique is not applicable.

Today, optimality assessment is popular in the computer sciences (Casavant & Kuhl, 1988; Raza, Zhu, & Chuah, 2011; Kim, 2013). Indeed, computer resources need to be assigned for tasks and reassigned as the priorities of tasks or parameters change with the goal of conserving computing resources. In Gounaris, Yfoulis, and Paton (2012), the purpose is not only to assign tasks to resources but to reassign tasks only when it is worth the disturbance. (Gounaris, Yfoulis, & Paton, 2012). Again, it considers a cost of moving from the current state to an optimum state but not the number of reassignments that are required or desired. This can be challenging if the cost or utility is difficult to measure; therefore, we propose a multiple objective optimization technique to determine the *n* best workload reassignments.

2.4 Product Test Scheduling

The fundamentals of our test scheduling problem resemble a vehicle routing problem (VRP) with a directed graph. The VRP was first introduced in 1959 by Dantzig and Ramser researching the optimal route for a fleet of gasoline delivery trucks with multiple gas station and one central depot (Dantzig & Ramser, 1959). Almost sixty years later the vehicle routing problem and variations of this problem are still of high interest due to its practical relevance.

A standard mathematical formulation of the integer programming model of a directed capacitated vehicle routing problem is provided in equations (5) – (10) (Laporte, Mercure, & Nobert, 1986; Toth & Vigo, 2001; Golden, Raghavan, Wasil, & ebrary, 2008; Toth & Vigo, 2014). Let the binary variable $x_{ij} = 1$ if a vehicle travels between *i* and *j*; 0 otherwise. The cost incurred if a vehicle travels between *i* and *j* is denoted by c_{ij}. Equation (5) is the objective to minimize the total routing costs. More specifically, the optimal solution identifies which vehicles travel which routes in the least cost manner. We do not use a cost parameter in the test scheduling problem; we concentrate on the total time to complete the five extreme environmental conditions tests.

Equations (6) and (7) work together to ensure the travel between *i* and *j* occurs only once. Equation (8) ensures that exactly $|K|$ routes are used from the depot, typically identifying the number of trucks available. Let *N* be the total number of stations that need to be serviced or, in more general terms, the total number of nodes that need to be visited. Equation (9) is a generalized constraint for subtours. We do not use this constraint in the test scheduling problem. Equation (10) requires the decision variable to be binary.

$$MIN \ \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \tag{5}$$

Subject To:

$$\sum_{j=1}^{m} x_{ij} = 1, \forall i \tag{6}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \forall j \tag{7}$$

$$\sum_{j=1}^{m} x_{0j} = |K| \tag{8}$$

$$\sum_{j=1}^{m} x_{ij} \geq r(S), \forall \ S \ subset \ of \ N \ where \ S \ cannot \ include \ the \ depot \tag{9}$$

$$x_{ij} \in \{0,1\}, \forall i, j \tag{10}$$

The test scheduling problem has five tests that each have established routes that must be traveled in that order by the same vehicle (or in our case, product); the tests can be completed in any order. The basic VRP mathematical model is not ideal for a test scheduling problem due to the addition of varying in-process and at-completion inspections by personnel. Fortunately, most VRP variants include parameters of travel time with time window constraints (VRPTW) (Toth & Vigo, 2001; 2014). This idea originates from the public transportation process of assigning different types of vehicles to multiple depots with timetabled trips that typically include fixed arrival and departure times at a minimal cost (Bunte & Kliewer, 2009).

Typically, equations (11) and (12) are added to the VRP integer programming model to accommodate the time constraints. The parameter, t_{ij}, identifies the time required to travel from node *i* to node *j* and time windows for each node $[a_i, b_i]$ are added. The variable T_i denotes the start time of service or visit for node *i*. With addition of subscript *k*, we may accommodate multiple trucks, although, we do not include this subscript due to the fact we only have one truck (or product) that needs to travel in the system. Equation (7) ensures the start time of service occurs in the given time window for node *i*. If the truck travels from node *i* to node *j*, equation (8) ensures that node *j* service starts after node *i* service is complete and the truck has traveled from node *i* to node *j* by using an arbitrarily large positive integer, *M*.

$$a_i \leq T_i \leq b_i, \forall i \tag{11}$$

$$T_i - T_j + M(x_{ij}) \leq M - t_{ij}, \forall i, j \tag{12}$$

The VRPTW mathematical structure is similar to our test scheduling problem; although, the order of the nodes is dependent on the fixed arrival and departure times of each node. In the test scheduling problem, identifying the start time to conduct each test is necessary but not dependent on a time window. As a result, we utilize a variation of equations (7) and (8) to accommodate our time requirements.

With the importance of safety and the need to reduce production cost, the automotive industry has been the focus of manufacturing test scheduling research (Chelst, Sidelko, Przebienda, Lockledge, & Mihailidis, 2001; Bartels & Zimmermann, 2009; Reich, Shi, Epelman, Cohn, Barnes, Arthurs, & Klampfl, 2016; Shi, Reich, Epelman, Klampfl, & Cohn, 2017). In 2009, Bartels and Zimmerman propose a mixed-integer linear programming model to evaluate the development costs in automotive R&D projects (Bartels & Zimmermann, 2009). More specifically, their objective is to minimize the number of experimental vehicles by optimizing the test schedule using a mixed-integer linear program formulation. There are a multitude of rules that need to be followed; for example, a subset of tests cannot be completed simultaneously or on the same vehicle. Bartels and Zimmerman's is industry specific and not readily generalizable to other applications.

More recently, Reich et al. use an integer program in combination with a column-generation algorithm to generate an automobile crash-test schedule for Ford (Reich et al., 2016). The primary objective is to maximize the utilization of vehicles for testing to minimize total cost due to the expensive nature of experimental vehicles. Each optimization evaluates a different sequence of tests and the impact on cost. In our paper, we strive to minimize time needed to complete all tests on one product whereas total time is a parameter in the Reich et al. configuration.

Additionally, the job shop scheduling problem (JSP) has many similar characteristics as the test scheduling problem. Basically, the JSP is the process of assigning n different jobs to m

different machines. The test scheduling problem is the process of assigning *n* different tests to specific time periods to ensure technician involvement during business operating hours. The number of machines required is not a factor in our problem. With that said, the number of similarities is too strong to neglect this area of research.

In the JSP, each job requires a sequence of operations and each operation must be completed for that job to be considered complete. Jobs are independent of each other and the processing times are fixed. The transportation time between jobs is included in the processing times. Only one job can be processed at a time on a machine. The operations and jobs can be processed in any order. In our test scheduling problem, the tests are jobs and the different testing time and inspections are operations; although, in the test scheduling problem the operations must be completed in order and at specific time periods. Identifying the optimal order and set of assignments while minimizing the make span is almost impossible due to the NP-hard nature of the combinatorial optimization JSP; therefore, genetic algorithms are used to solve these problems to optimality (Garey, Johnson, & Sethi, 1976; Cheng, Gen, & Tsujimura, 1996; 1999; Potts & Strusevich, 2009).

Cheng, Gen and Tsujimura identified a list of nine categories for genetic algorithms used in attempts to solve the JSP. Unfortunately, all categories utilize multiple machines and the basic assignment problem of job order does not factor in technician involvement. One category that resembles our test scheduling problem is called priority rule-based representation. Here, job order is decided based on priority dispatch rules. Dispatch rules help decide which operation is next in line for completion. Rules can incorporate the shortest processing time, longest processing time, earliest due date, first come first serve, or even random assignment. To date, business operating hours and technician involvement has not been considered in this line of research; therefore, the genetic algorithms to date for the JSP used do not apply to the test scheduling problem considered in this dissertation.

Chapter 2 : Identifying the Best Path to Optimality in a Grants Administration Workload Assignment Problem

Identifying the Best Path to Optimality in a Grants Administration Workload Assignment Problem

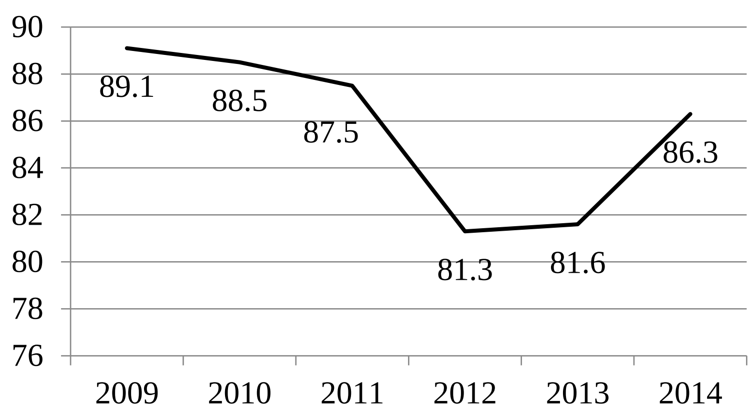
ABSTRACT

This research addresses a grant administration workload-to-staff assignment problem that occurs in the Office of Research and Sponsored Programs (ORSP) at land-grant universities in the pre-award process. We first identify the optimal (utopian) workload assignment plan using a mixed-integer linear programming problem. The optimal assignment of staff members to workload (academic departments) may differ considerably from the status quo, requiring multiple reassignments from the current state to reach optimality. The number of reassignments raises concerns related to loss of administrator-department relationship, loss of department-related knowledge, and increase in managerial inconvenience. To achieve the best workload reassignment with the fewest changes from the current status quo, while still placing a greater emphasis on the effective use of limited resources, we propose and illustrate a multiple objective optimization technique to identify the *n* best departmental reassignments from the current state that provide the greatest progress toward the utopian balanced workload solution. Solving this problem over several values of *n* and plotting the results allows the decision maker to visualize the trade-off between the number of reassignments and the resulting progress achieved toward the utopian solution. This system supports the ORSP pre-award administrator in making an informed decision about the best number of reassignments (*n*) to choose based on an objective assessment of the relevant trade-offs.

1.0 Introduction

Over the past decade, many public universities faced significant financial challenges due to the unabated reduction in state funding. The State Higher Education Finance Report showed a 7 percent reduction in state support for higher education from 2011 to 2012 representing a $6.2 billion decrease with a slight increase of 0.4% ($300 million) in 2013. (State Higher Education Executive Officers, 2011; 2012; 2013; 2014). More recently, state and local funding increased 5.7% in 2014 reaching $86.3 billion, still short of 2009-2011 levels. Figure 2-1 summarizes the downward trend in state and local funding for U.S. public higher education from 2009 to 2014 (State Higher Education Executive Officers, 2011; 2012; 2013; 2014). Historically, public universities could transfer budget shortfalls onto the student populace via increased tuition. However, rising negative public sentiment towards university tuition inflation along with the associated ballooning student debt crisis has increased price sensitivity and amplified federal government pressure to curb rising tuition costs (Lewin, 2011; Martin, 2012). From 2008 to 2013, full-time equivalent educational revenues including tuition increases have dropped on average 5.9 percent in the U.S. (State Higher Education Executive Officers, 2014).

Figure 2-1: State and Local Funding for Higher Education from 2009 to 2014



While universities have attempted to maintain a sustainable level of revenue to cover operating costs, this has proved difficult due to the fixed nature of most university operating costs coupled with the inability to significantly increase tuition rates to cover funding shortfalls. Therefore, an increasing reliance on government and corporate funding through sponsored grants and research programs has become a mainstay for most research universities.

Additionally, academic prestige and research ranking by government agencies, like the National Science Foundation (NSF), are derived from a university's ability to maintain and increase funding levels through research projects. According to the National Science Foundation's Higher Education Research and Development Survey, university spending on research and development grew 6.3 percent from 2010 to 2011, reaching the highest level ever of $65 billion (Britt, 2012).

In 2008, the federal government provided 57 percent of the basic research funding in the U.S. (Sargent Jr., Esworthy, Matthews, Morgan, Moteff, Schacht, Smith, & Upton, 2011; Sargent Jr., Esworthy, Gonzalez, Matthews, Morgan, Moteff, Schacht, Smith, & Upton, 2012), intensifying the workload on potential researchers to search out and apply for research funding opportunities. As one might expect, applying for federal and corporate research funding can require the completion and submission of a myriad of detailed government mandated forms. The researcher must subsequently provide well-documented results derived from the funded research.

The last decade has been a significantly dynamic period for universities, especially in the area of sponsored research. The changing environment has resulted in the following pressing issues that affect research administration and university research: increased competition with other universities and industry partners for research funds, increased regulatory compliance and accountability practices, and significant budget cuts from states (Casey Jr., 2005; Shelton, 2009; Chun, 2010). Additionally, increasing regulatory compliance and accountability practices requires researchers to spend more time on administrative duties and less time on research (Kirby, 1996; Rockwell, 2009; Stanley Jr. & McCartney, 2009; Board on Higher Education and Workforce Policy and Global Affairs: Committee on Research Universities, 2012). As a result, offices of research and sponsored program administration (ORSP) have been setup to handle many such administrative tasks.

Most universities have an Office of Research and Sponsored Programs that assists researchers with the management of the application, review, award, and administration processes of sponsored research (Kirby, 1996; Saha, Ahmed, & Hanumandla, 2011). The National Academy of Science stresses the importance of efficient administration of university research due to concerns about increased administrative and regulatory-related reporting requirements (Board on Higher Education and Workforce Policy and Global Affairs: Committee on Research Universities, 2012). In addition, recent research published by the National Council of University Research Administrators highlights the importance of a good work environment to improve quality and performance of ORSP staff (Saha et al., 2011). Given the significance of employee workload as a key determinant of employee satisfaction (Saha et al., 2011) and efficient administration, it is critical to effectively assign workloads to ORSP staff members so as to minimize the use of overtime and also balance it equitably among ORSP workers.

This research addresses a recurring decision problem that occurs in the ORSP at a large, public, land-grant university in the pre-award process, which includes the application and review portion of sponsored research. We first identify the optimal (utopian) workload assignment plan using a mixed-integer linear programming problem. Unfortunately, the optimal assignment of

ORSP staff members to academic departments where the workload is located in this problem differs considerably from the status quo, requiring multiple reassignments from the current state to reach optimality. The number of reassignments creates concerns related to loss of administrator-department relationship, loss of department-related knowledge, and increase in managerial inconvenience.

To achieve the best workload reassignment with the fewest changes from the current status quo, while still placing a greater emphasis on the effective use of limited resources, we propose and illustrate a multiple objective optimization technique to identify the *n* best departmental reassignments from the current state that provide the greatest progress toward the utopian solution. Solving this problem over several values of *n* and plotting the results allows the decision maker to visualize the trade-off between the number of reassignments and the resulting progress achieved toward the utopian solution. This system supports the ORSP pre-award administrator in making an informed decision about the best number of reassignments (*n*) to choose based on an objective assessment of the relevant trade-offs.

The remainder of this paper is organized as follows. In section 2, a literature review evaluating task assignment, load balancing and optimality assessment is provided. Section 3 examines our methodology and techniques used to assess the closeness to optimality in relation to the number of reassignments desired. Section 4 provides a demonstration of the mathematical model and the findings from that implementation followed by conclusions and areas for future work in section 5.

2.0 Literature Review

The nature of the work in the present paper leads to an evaluation of literature in task assignment, load balancing and optimality assessment. A brief overview of these research fields is provided to support the methodology and techniques used to develop and assess the trade-off between the number of reassignments and proximity to optimality.

2.1 Task Assignment

In the 1950's, the simple assignment problem assessed a set of *n* personnel, *n* job tasks, and individual task qualifications to identify the best set of personnel-to-task assignments that maximizes the number of tasks completed while allowing only one task to one person (Votaw Jr. & Orden, 1951; Kuhn, 1955). The personnel-task qualifications were identified in a binary fashion with a 1 if the person was qualified to complete the task and 0 otherwise. In comparison, the general assignment problem enhanced the simple assignment problem by creating a rating matrix rather than the personnel-task qualifications and strived to maximize the sum of the ratings (Votaw Jr. & Orden, 1951; Kuhn, 1955). The rating matrix utilized positive integers to rate the personnel based on the task rather than a 0-1 classification.

In 1951, Votaw and Orden presented several methods to address the general assignment problem, primarily concentrating on the simplex method, with satisfactory solutions but computational capacity constraints (Votaw Jr. & Orden, 1951). Later, Kuhn proposed a polynomial algorithm known as the Hungarian method to solve the general assignment problem in an effort to improve the computational constraints (Kuhn, 1955; 1956).

Later, the general assignment problem evolved into the generalized assignment problem (GAP) by introducing a cost matrix, allowing multiple tasks to one person, and providing the best set of personnel-to-task assignments based on the minimum total cost (Ross & Soland, 1975; Fisher et al., 1986). Let the binary variable $x_{ij} = 1$ if person *i* is assigned to task *j*; and 0 otherwise. The cost incurred if person *i* is assigned to task *j* is denoted by c_{ij}. The resources required by agent *i* to complete task *j* is denoted r_{ij} and the total amount of resource available for person *i* is identified by b_i.The standard formulation of a GAP is the following (Ross & Soland, 1975; Oncan, 2007). Note, the task assignment problem is formulated identical to the GAP with the addition of communication cost in the objective function (Li et al., 2013).

$$\text{MIN} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

Subject To:

$$\sum_{j=1}^{n} r_{ij} x_{ij} \leq b_i, i = 1, \ldots, m \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1,\ j = 1, \ldots, n \tag{3a}$$

$$x_{ij} \in \{0,1\}, i = 1, \ldots, n \text{ and } j = 1, \ldots, m \tag{4}$$

Given the nature of the GAP being an NP-hard combinatorial optimization problem, Dakin (1965) suggested a tree-search algorithm motivated by Land and Doig (1960) to address such a problem (Land & Doig, 1960; Dakin, 1965; Fisher et al., 1986). Later, Ross and Soland (1975) expand on the tree-search method and present the branch and bound method to improve computational efficiency for the GAP.

Notably, personnel-to-task assignments can become a complex resource allocation challenge based on the problem size and integrality requirements. In response, Cattrysse and Van Wassenhove (1992) provided a review of algorithms that address the GAP utilizing different relaxation techniques developed in the 1970's and 1980's (Cattrysse & Van Wassenhove, 1992). In 2007, Oncan provided a survey of the GAP along with current solution procedures including enhanced relaxation techniques in addition to heuristic and metaheuristic procedures (Oncan, 2007). Once again, relaxation techniques and heuristic procedures are used in an attempt to improve computational time, although, integer optimal solutions may be difficult to identify.

In addition to various methods to solve the GAP, variations or extensions to the mathematical structure of the GAP have been presented. In 1998, Park et. al presented the generalized multi-assignment problem by altering equation (3a) (Park et al., 1998).

$$\sum_{i=1}^{m} x_{ij} \geq a_j \quad j = 1, \dots, n \qquad (3b)$$

This formulation allows multiple agents to be assigned to a single task. In the event $a_j = 1$ for all j, the formulation becomes the GAP. Our proposed method utilizes a similar alteration to the GAP by allowing a set number of agents to be assigned to a single task.

Later, Naus (2004) presented the elastic generalized assignment problem which allows the agents to exceed or violate the resource capacity in equation (2) by using over time and under time. Alterations to equations (1) and (2) must be made to introduce the slack and surplus variables with the additional cost of over time or under time. Our proposed method allows for over time and under time without factoring in the cost or utility.

In 2013, Li et. al. suggests a logarithmic approach to the GAP to reduce the number of binary variables and inequality constraints. This variation is used to improve computational time in large-sized GAPs. To maintain simplicity, we concentrate on the mathematical structure of the GAP demonstrated in Equations (1), (2), (3b) and (4).

2.2 Load Balancing

In 2006, Harvey et. al. evaluated the load balancing problem which is an extension of the GAP (Harvey et al., 2006). Typically, the term load balancing is used in the context of computer processors, distributed systems, and project assignment (Chou & Abraham, 1982; Zhou, 1988; Liang et al., 2010; Penmatsa & Chronopoulos, 2011; Ali & Khan, 2012; Neelakantan, 2012). This research concentrates on the idea of reallocating workload from nodes exceeding capacity to idle or under-utilized nodes to meet or exceed specific performance criteria.

Line balancing is related to load balancing in terms of distributing the work across the workstations equally; however, line balancing also emphasizes workstation precedence constraints (Casavant & Kuhl, 1988; Becker & Scholl, 2006; Gen et al., 2008). The line balancing problem and its underlying problem of task assignment are well-known as it has long been a troubling issue in manufacturing processes. Line Balancing, also known as assembly line balancing, dates back to automobile manufacturing with Henry Ford in 1913 (Becker & Scholl, 2006; Gen et al., 2008). Specifically, tasks are assigned to workstations in a sequential order on an assembly line to meet one or more objectives.

Salveson formulated the first mathematical line balancing problem in a manufacturing context in 1954 (Ghosh & Gagnon, 1989; Battaia & Dolgui, 2013). Since then, more modern manufacturing systems appeared with disassembly, parallel lines, or workstations operated by computer-controlled robots (Becker & Scholl, 2006; Battaia & Dolgui, 2013). This has kept the line balancing problem of designing or re-designing the line of current interest to both researchers and practitioners.

2.3 Optimality Assessment

The assignment and workload balancing algorithms have one point in common; they seek an optimal solution. At times, knowing what is optimal is not sufficient and decision makers are interested in exploring suboptimal solutions due to other issues associated with application, implementation, and unarticulated objectives. Most assessments of suboptimal versus optimal solutions are focused on a cost or utility, not the actual number of reassignments. This can be challenging if the cost or utility is difficult to measure, therefore, we propose a multiple objective optimization technique to determine the *n* best workload reassignments.

In Murty's 1968 article, a partitioning technique is used to rank the *k* best assignments in order of increasing cost (Murty, 1968). Although the research is 45 years old, the technique is still used today. In 2008, small modifications of Murty's technique led to the development of an algorithm with a sub-optimal starting point and a search method for the best improvement rather than the lowest degradation (Pedersen et al., 2008). However, that would give us the *k*-best solutions, which would not inform us of the number of reassignments required.

In 2004, Zulch, Rottinger, and Vollstedt acknowledged the issue of reassigning personnel to tasks, specifically in a manufacturing environment (Zülch et al., 2004). The goal is to reassign personnel to improve the overall utility which includes average lead time, system output, average workload, and labor costs; however, the individual cost of reassignment is not evaluated. A few years later, Gamberini, Grassi, and Rimini evaluate the cost of reassignment when rebalancing of a production line is needed due to changes in parameters (Gamberini et al., 2006). In the model, the rebalance of the line creates a solution that optimizes the line and reduces the cost of reassignment. This logic may reduce the number of reassignments realized, but the decision is not based on the number of reassignments desired. If the cost of reassignment is not known, the technique is not applicable.

Today, workload balancing and optimality assessment is popular in the computer sciences (Casavant & Kuhl, 1988; Raza et al., 2011; Kim, 2013). Indeed, computer resources need to be assigned for tasks and reassigned (or rebalanced) as the priorities of tasks or parameters change with the goal of sparing computing resources. In Gounaris, Yfoulis, and Paton 2012 article, the purpose is not only to assign tasks to resources but to reassign tasks only when it is worth the disturbance (Gounaris et al., 2012). Again, it considers a cost of moving from the current state to an optimum state but not the number of reassignments that are required or desired.

3.0 Mathematical Model

We first address the problem of identifying the optimal (utopian) set of assignments of ORSP staff members with varying resource availabilities to academic departments with varying support requirements. We assume that each ORSP staff member may be assigned to a maximum of δ departments. Additionally, each academic department must be assigned one staff member unless the total workload for one department exceeds what one full-time staff member can accommodate. These assignments are made based on estimated department workload hours, ε_j, and ORSP staff member availability of work hours, γ_i. Department workload, ε_j, is estimated by a projected number of proposals each year where each proposal requires a specific amount of work hours. Projecting the number of proposals and hours per proposal is currently handled by ORSP administrators and is outside the scope of this research. However, these projections are a necessary data input for our proposed methodology.

Each ORSP staff member's available number of annual work hours, γ_i, is determined by his/her employment status (e.g. full-time or part time) and may vary from employee to employee. The parameter τ is the available number of work hours for one full-time employee in a year.

3.1 A Model for Optimal ORSP Scheduling

Let the binary variable $x_{ij} = 1$ if staff member *i* is assigned to academic department *j*; and 0 otherwise. Let the variable d_{ij} denote the number of workload hours assigned to ORSP staff member *i* from academic department *j*. Let the surplus variable o_i denote the number of assigned hours over the available amount for ORSP staff member *i* and the slack variable u_{i} denote the number of hours under the available amount for ORSP staff member *i*. Finally, let the variable B denote the maximum individual deviation for all ORSP staff members from their available capacities, γ_i.

A mixed-integer linear programming (MILP) model for this problem is given below in equations (5) - (13). Equations (5), (6), and (7) work in consort to balance the workload among the ORSP employees by minimizing the maximum amount (B) by which any employee's actual assigned hours (d_{ij}) differs from his or her available number of hours (γ_i). In equation (6), the slack and surplus variables (u_i and o_i) measure the deviation between employee *i*'s assigned hours (d_{ij}) and available hours (γ_i). Equations (5) and (7) then minimize the maximum of these deviations.

$$MIN \text{ B} \tag{5}$$

Subject To:

$$\sum_{j=1}^{m} d_{ij} - o_i + u_i = \gamma_i, \forall i \tag{6}$$

$$o_i + u_i - \mathrm{B} \leq 0, \forall i \tag{7}$$

$$\sum_{i=1}^{n} d_{ij} = \varepsilon_j, \forall j \tag{8}$$

$$\sum_{i=1}^{n} x_{ij} = \mathrm{MAX}(1, \left\lceil \frac{\varepsilon_j}{\tau} \right\rceil), \forall j \tag{9}$$

$$\sum_{j=1}^{m} x_{ij} \leq \delta, \forall i \tag{10}$$

$$\gamma_i x_{ij} - d_{ij} \geq 0, \forall i, j \tag{11}$$

$$x_{ij} \in \{0,1\}, \forall i, j \tag{12}$$

$$d_{ij}, o_i, u_i, \mathrm{B} \geq 0 \tag{13}$$

Equation (8) ensures that for each academic department *j* the total amount of workload assigned equals the total amount required. Equation (9) requires each academic department *j* be assigned to only one ORSP staff member, unless the total amount of workload for one department exceeds τ, (or what one full-time staff member can provide). In the event the workload for a department exceeds an even multiple of τ, the ceiling operator in (9) increases the right-hand side value of this constraint to the next highest integer value. Equation (10) ensures each ORSP staff member *i* is assigned to a maximum of δ departments. Equation (11) enforces the logical relationship between the decision variables x_{ij} and d_{ij}. Specifically, constraint (11) ensures ORSP staff member *i* is assigned to academic department *j* if an amount of workload from department *j* is assigned to staff member *i*.

Equation (12) requires the assignment decision variables x_{ij} to be binary. Equation (13) imposes non-negativity constraints for the remaining decision variables. In the event an ORSP manager would like to ensure a particular staff member *i* be assigned to a particular department *j* a simple constraint can be added for the corresponding *i* and *j* values: $x_{ij} = 1$. Similarly, a constraint can be added to ensure a particular staff member *i* is not assigned to a particular department *j* by setting $x_{ij} = 0$.

3.2 Modifying the Model for Tradeoffs

The solution to the MILP in (5) - (13) provides an optimal set of work assignments in (as shown later) a reasonable amount of CPU time. However, the "optimal" solution may differ considerably from the status quo, and require multiple reassignments from the current state to implement. Thus, the optimal solution obtained in this manner is somewhat utopian in that it ignores the relational (and other) costs associated with reassigning ORSP staff from one department to another. That is, the MILP model in (5) - (13) is indifferent between an assignment that achieves the utopian solution's optimal objective function value in 25 reassignments and one that achieves the same utopian objective function value in, say, 10 reassignments. However, the ORSP administrator would likely prefer to make 10 reassignments rather than 25. Furthermore, a solution involving 5 reassignments that achieves 98% of the utopian objective function value might be preferable than the one involving 10 reassignments.

From this discussion, we are motivated to propose a method to determine the *n* best workload reassignments. The MILP model for this problem (for a given value of *n*) is a simple extension of the previous model, with the addition of equation (14) where x^*_{ij} denotes the set of current state variables that are equal to 1 (or, in other words, the set of current ORSP staff assignments), and λ denotes an integer value representing the number of allowable reassignments.

$$\sum x^*_{ij} = \lambda \ \forall \ i, j \in \Omega, \text{ where } \lambda \text{ is Integer and } \Omega = \left\{i, j \;\middle|\; x^*_{ij}\right\} \quad (14)$$

Equation (14) is based on the decision maker's selection of the number of reassignments desirable from the current state. The integer value, λ, is calculated by identifying the sum of current state variables and subtracting the number of desired reassignments. This enables the decision maker to assess the number of reassignments in comparison to the resulting solutions' proximity to the utopian solution.

4.0 Application

We demonstrated the value of our MILP method for evaluating the trade-off between the number of reassignments and the resulting progress achieved toward optimality, we utilized data from an American university that engages in extensive funded research. In 2014, this university was ranked in the top 40 universities for research and development expenditures by the National Science Foundation (National Center for Science and Engineering Statistics, 2015).

With the increased focus on research dollars, the American university's ORSP is forced to complete more administrative work with less resources (Sedwick, 2009) ; the primary resource constraint being staff member availability of work hours (Magruder, 2013). The ORSP management team struggles with making the assignment of academic departments to staff members while keeping the workload as balanced as possible. Currently, the ORSP management team uses their best estimate of what is equal or balanced based on individual experience and personal judgment.

Typically, the ORSP team must rearrange the department-to-staff member assignments numerous times throughout the year to keep the workload balanced and staff members content. Unfortunately, the number one complaint by faculty is the constant shift of pre-award administrators (Magruder, 2013). This frustration stems from the faculty's additional time required to initiate, build, and maintain the relationship with the new pre-award administrator. By implementing our MILP method, the ORSP management team can evaluate the trade-off between the number of reassignments and the resulting progress achieved toward the optimal, balanced set of department-to-staff member assignments.

4.1 Identification of Optimal Solution

We identify the optimal set of assignments of ORSP staff to academic departments prior to analyzing the trade-off. In this application, we have 8 scenarios where a trade-off assessment will be demonstrated. Table 2-1 shows the breakdown of each scenario with regard to the release date of the department-to-staff assignments, number of administrators available including how many are full-time versus part time, and the number of departments. In addition, we assume each ORSP staff member can be assigned a maximum of 10 (δ) departments.

Table 2-1: ORSP Optimality Trade-Off Scenarios

Scenario	ORSP Release Date	# of Admins	Dummy Variable	Administrators FT/.25PT/.5PT/.75 PT	# of Departments
A	January 2011	13	Yes	8/2/1/2	88
B	March 2011	13	Yes	8/2/1/2	88
C	November 2011	15	Yes	10/2/2/1	88
D	May 2012	14	Yes	10/2/2/0	88
E	July 2012	14	Yes	10/2/2/0	88
F	December 2012	13	No	9/2/2/0	88
G	March 2013	13	No	9/2/2/0	88
H	July 2013	10	No	7/2/1/0	88

We assume 40-hour work weeks with 47 work weeks in a year (52 weeks in a year minus 2 weeks in December and 3 weeks from January-November for vacation and sick time = 47 weeks) totaling 1880 (τ) potential work hours available for one full-time staff member per year. The total available work hours are adjusted for different types of employment status where full-time staff members are given an availability value of 100% and part time staff members can be given a value of 25%, 50% or 75%. For example, a part time staff member with a value of 50% will be available 0.5 x 1880 hours totaling 940 hours for the year. Therefore, the parameter γ_i can vary but not exceed 1880 hours.

In some instances, the ORSP did not make a work assignment for an academic department; as a result, we created a dummy administrator variable and assigned the available departments to that dummy variable. As noted earlier, equation (9) in the utopian model requires each academic department be assigned a minimum of one ORSP staff member. As a result, the dummy variable is assigned 0 hours of availability, or 0% to ensure each department is assigned an actual ORSP administrator.

Each academic department's workload (ε_j) is forecasted by evaluating the year-to-year growth of total number of grant proposals from 2000-2013 showing an average of 1 percent growth. The average level of growth is used to calculate the expected number of proposals for each department in the coming year. We assume the workload per proposal may vary between 4 hours and 12 hours. In our computational testing, the amount of work hours for each proposal will be calculated randomly within this range.

The mixed-integer linear programming model for this problem is optimized using IBM ILOG CPLEX Optimizer 12.5. Table 2-2 shows each scenario's model characteristics including the number of binary variables, non-negative variables, and constraints.

Table 2-3 shows the utopian objective value for each scenario. Each scenario's objective value is calculated within 0.5 percent of optimality or 2 hours of CPU time. Recall that the objective value is the amount of workload the ORSP employee is overworked in a specific scenario’s timeframe. The utopian objective value minimizes the maximum amount of B. The utopian objective value is substantially less that the ORSP objective value, demonstrating the need to balance workload. See Table 2-3 for a more detailed breakdown of each scenario.

Table 2-2: ORSP Optimality Model Characteristics by Scenario

Scenario	# of Binary Variables	# of Non-Negative Variables	# of Constraints
A	1144	1171	1450
B	1144	1171	1450
C	1320	1351	1632
D	1232	1261	1541
E	1232	1261	1541
F	1144	1171	1450
G	1144	1171	1450
H	880	901	1177

Table 2-3: ORSP Optimality Model Objective Value Results versus Actual

Scenario	Percent to Optimality	CPU Time (seconds)	Utopian Overwork (hours	ORSP Overwork (hours)
A	98.8%	7200	62	459
B	99.6%	7200	160	820
C	97.3%	7200	49	412
D	91.4%	7200	24	165
E	94.0%	7200	26	419
F	100.0%	0.23	117	304
G	78.0%	7200	7	254
H	99.2%	7200	97	966

Table 2-4 shows the number of reassignments for the utopian and actual solutions for each scenario. Unfortunately, the optimal assignment for each scenario requires a large number of reassignments from the current state to reach optimality. As previously stated, the number of reassignments creates concerns related to loss of administrator-department relationship, loss of department-related knowledge, and increase in managerial inconvenience. Therefore, the closeness to optimality assessment is necessary to reduce disruption.

Table 2-4: ORSP Optimality Model Reassignment Results versus Actual

Scenario	# Reassignments (Utopian)	# Reassignments (Actual)	ORSP Initial State
A	169	59	September 2010
B	159	5	January 2011
C	172	105	March 2011
D	157	18	November 2011
E	165	21	May 2012
F	163	30	July 2012
G	160	6	December 2012
H	170	53	March 2013

4.2 Closeness to Optimality Assessment

The current set of assignments of the ORSP staff was identified for each scenario using the organizational staff contacts data from the university's ORSP. The established current state variables (x_{ij}^*) were set equal to 1. Using the mixed-integer linear programming model for evaluating the closeness to optimality based on number of reassignments, the tradeoff assessment is identified. We optimized each scenario for various levels of λ. In other words, we allowed 0 to 20 reassignments to be made from the current state. Each value of λ requires the model to be optimized separately.

The mixed-integer linear programming model for this problem is optimized using IBM ILOG CPLEX Optimizer 12.5. In table 2-5 each model required a certain number of binary variables, non-negative variables, and constraints. Each scenario's tradeoff assessment is calculated within 0.5 percent of optimality or 30 minutes (1800 seconds) of CPU time

Table 2-5: ORSP Optimality Trade-off Model Characteristics by Scenario

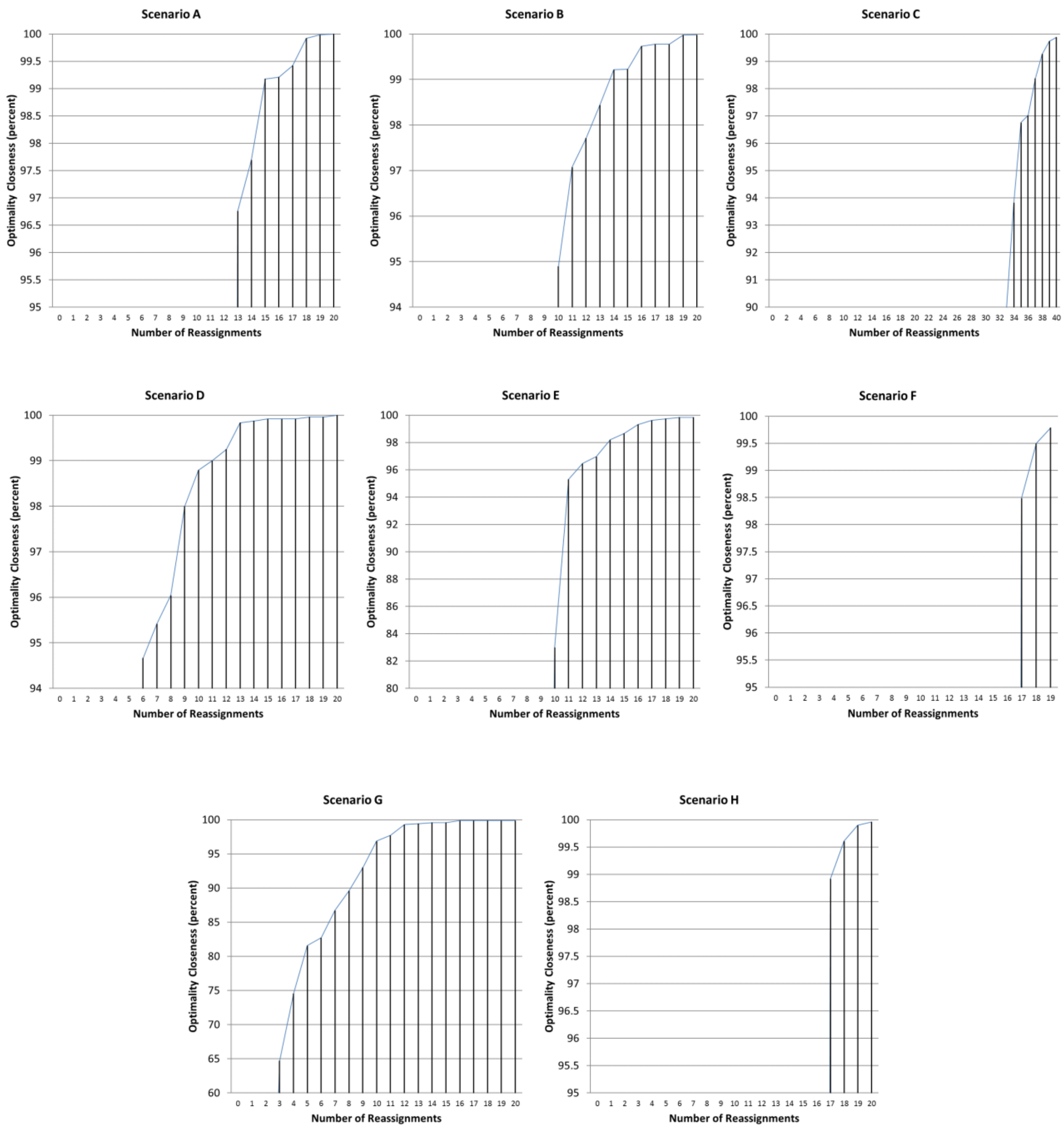
Scenario	# of Binary Variables	# of Non-Negative Variables	# of Constraints
A	1144	1171	1451
B	1144	1171	1451
C	1320	1351	1633
D	1232	1261	1542
E	1232	1261	1542
F	1144	1171	1451
G	1144	1171	1451
H	880	901	1178

All scenarios returned infeasible solutions for certain number of allowable reassignments from the current state, demonstrating that the current state did not meet the constraints required to perform adequately given the resources. Scenarios A and D reach 100% of optimality when we allowed 20 reassignments. Scenario B, E, F, G and H reach 99.9% of optimality when we allowed 20 reassignments. At this point, each additional reassignment provides a small increase in closeness to optimality without an extreme amount of CPU time. Scenario C returned infeasible solutions for 0 to 32 allowable reassignments from the current state, therefore, we made an exception and allowed 40 reassignments. We reach 99.9% of optimality when we allowed 40 reassignments. See Table 2-6 for a summary of results for each scenario. See Figure 2-2 for a tradeoff assessment by scenario.

Table 2-6 : ORSP Results Summary for Trade-off Assessment by Scenario

Scenario	# of Reassignments to Feasible Solution	Objective Value	% toward Optimality	CPU Time (seconds)	Objective Value @ 20 Reassignments	% toward Optimality	CPU Time (seconds)
A	13	263	96.8	0.03	62	100	279.16
B	10	977	94.9	0.06	163	99.9	1800
C	33	539	90	0.11	55	99.9	1800
D	6	152	94.7	0.3	24	100	1800
E	10	468	83	0.14	30	99.9	1800
F	17	294	98.5	0.11	125	99.9	0.58
G	3	64.7	254	0.16	8	99.9	1800
H	17	98.9	202	0.13	101	99.9	4.03

Figure 2-2 : Trade-off Assessment by Scenario: Closeness to Optimality versus Number of Reassignments



5.0 Conclusion

This research addresses the pre-award workload-to-staff assignment of sponsored research in land grant universities. We identify the optimal workload assignment plan using a mixed-integer linear programming problem, although, the optimal assignment requires multiple reassignments from the current state to reach optimality.

The number of reassignments creates concern; therefore, we propose a technique to identify the *n* best departmental reassignments from the current state that provide the greatest progress toward the optimal solution. The decision maker can visualize the trade-off between the number of reassignments and the resulting progress achieved toward the optimal solution and make an informed personnel scheduling decision.

Chapter 3 : The Best Path to Optimality for a Gymnastics Facility Task Assignment and Tour Scheduling Problem

The Best Path to Optimality for a Gymnastics Facility Task Assignment and Tour Scheduling Problem

ABSTRACT

With the excitement of the 2016 Olympics, the popularity of the ancient Greek sport, gymnastics, is on the rise. Gymnastics has become one of the leading summer Olympic sports to watch and children around the world are getting involved. As a result, gymnastics facilities must identify a weekly personnel schedule that seeks the most cost-effective set of coach-to-program assignments using a cross-trained workforce with varying availability. Typically, gymnastic facility business owners require 24+ hours to complete the formulation of an 8-week personnel schedule, which is not easily adjusted due to the complexity of hard copy calculations, employee job qualifications and established coach-to-gymnast relationships (Kennedy, 2017; Tessier, 2017). We first identify the optimal (utopian) assignment plan using an integer linear programming problem. The optimal coach-to-program assignment can differ considerably from the status quo, requiring multiple reassignments from the current state to reach optimality. Making these adjustments with minimal coach reassignment maintains a level of comfort and familiarity with the children and sustains a level of staffing cost which is ideal for the business operations. We utilize the multiple objective optimization technique to identify the *n* best coach-to-program reassignments from the current state that provide the greatest progress toward the utopian staffing cost solution. This provides the decision maker with a summary of the trade-off between the number of program reassignments and the resulting progress achieved toward the optimal staffing cost. The decision maker can then make an informed decision about the best number of program reassignments (*n*) to choose based on the staffing cost objective; in other

words, identifying the desired level of disruption (or reassignment) and the impact on staffing cost.

1.0 Introduction

Following the excitement and anticipation of the 2016 Olympic Games, the popularity of the ancient Greek sport, gymnastics, is once again on the rise. Gymnastics has become one of the leading summer Olympic sports to watch and children around the world are getting involved. In 2012, USA Gymnastics conducted a survey of over 335 facilities and approximately 67% report growth in enrollment and 18% are holding steady; in addition, 88% expected the impact of the 2012 Olympic Games to increase enrollment (Holcomb, 2012).

This paper considers the unique challenges of personnel scheduling within one of the largest gymnastics facilities in the U.S. Every 8 weeks the business owner must schedule a sufficient amount of coaches for each shift for each day along with the program assignment over a 7-day operation while ensuring all employees scheduled are qualified to teach that program. In other words, gymnastics facilities need a weekly personnel schedule that identifies the most cost-effective set of assignments using a cross-trained workforce with varying availability.

Typically, gymnastic facility business owners require 24+ hours to complete the formulation of an 8-week personnel schedule which is not easily adjusted due to the complexity of hard copy calculations, employee job qualifications and established coach-to-gymnast relationships (Kennedy, 2017; Tessier, 2017). We first identify the optimal (utopian) coach-to-program assignment plan using an integer linear programming problem.

Unfortunately, the optimal set of coach-to-program assignments differs considerably from the status quo, requiring multiple reassignments from the current state to reach optimality. Making these adjustments with minimal coach reassignment maintains a level of comfort and familiarity with the children and sustains a level of staffing cost which is ideal for the business.

To achieve the most desired coach-to-program reassignment plan, we utilize the multiple objective optimization technique to identify the *n* best coach-to-program reassignments from the current state that provide the greatest progress toward the utopian staffing cost solution. We provide the decision maker with a summary of the trade-off between the number of program reassignments and the resulting progress achieved toward the optimal staffing cost. The decision maker can then make an informed decision about the best number of program reassignments (*n*) to choose based on the staffing cost objective; in other words, identifying the level of disruption (or reassignment) and staffing cost desired.

The remainder of this paper is organized as follows. In section 2, a literature review evaluating task assignment and tour scheduling, cross-trained workforces and optimality assessment is provided. Section 3 examines our methodology and techniques used to assess the closeness to optimality in relation to the number of reassignments desired. Section 4 provides a demonstration of the mathematical model and the findings from that implementation followed by conclusions and areas for future work in section 5.

2.0 Literature Review

The problem addressed in this paper is an integer personnel scheduling model that optimizes staffing cost by identifying the ideal coach-to-program set of assignments while incorporating employee availability and employee qualifications over a 7-day operation. A brief overview of the key contributions to the literature in task assignment and tour scheduling, cross-trained workforces and optimality assessment is provided to support the methodology and techniques used to develop the multiple objective optimization technique that identifies the reassignment to objective tradeoff.

2.1 Task Assignment and Tour Scheduling

This research addresses a coach-to-program task assignment and tour scheduling problem using a cross-trained workforce within a gymnastics facility. Dantzig and Edie were the first to use probability theory and linear programming to address the task assignment problem (Dantzig, 1954; Edie, 1954). Later, Kuhn proposed the well-known Hungarian polynomial algorithm to solve the assignment problem (Kuhn, 1955; 1956).

The general assignment problem assessed a set of n personnel, n job tasks, and individual task qualifications rating matrix to identify the best set of personnel-to-task assignments that maximized the sum of the ratings while allowing only one task to one person (Votaw Jr. & Orden, 1951; Kuhn, 1955). Later, the general assignment problem evolved into the generalized assignment problem (GAP) by introducing a cost matrix, allowing multiple tasks to one person, and providing the best set of personnel-to-task assignments based on the minimum total cost (Ross & Soland, 1975; Fisher et al., 1986).

The standard formulation of a GAP is the following (Ross & Soland, 1975; Oncan, 2007). Let the binary variable $x_{ij} = 1$ if person *i* is assigned to task *j*; and 0 otherwise. The cost incurred if person *i* is assigned to task *j* is denoted by c_{ij}. The resources required by agent *i* to complete task *j* is denoted r_{ij} and the total amount of resource available for person *i* is identified by b_i. Note, the task assignment problem is formulated identical to the GAP with the addition of communication cost in the objective function (Li et al., 2013).

$$\text{MIN} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

Subject To:

$$\sum_{j=1}^{n} r_{ij}x_{ij} \leq b_i, i = 1, \dots, m \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1,\ j = 1, \dots, n \tag{3a}$$

$$x_{ij} \in \{0,1\}, i = 1, \dots, n \ and \ j = 1, \dots, m \tag{4}$$

In addition to various methods to solve the GAP, variations or extensions to the mathematical structure of the GAP have been presented. In 1998, Park et. al presented the *generalized multi-assignment problem* by altering equation (3a) (Park et al., 1998). This formulation allows multiple agents to be assigned to a single task. In the event $a_j = 1$ for all j, the formulation becomes the GAP.

$$\sum_{i=1}^{m} x_{ij} \geq a_j \ \ j = 1, \dots, n \tag{3b}$$

Given the nature of the GAP being an NP-hard combinatorial optimization problem, Dakin suggested a tree-search algorithm in 1965 motivated by Land and Doig to address such a problem (Land & Doig, 1960; Dakin, 1965; Fisher et al., 1986). Later, Ross and Soland expand on the tree-search method and present the branch and bound method to improve computational efficiency for the GAP (Ross & Soland, 1975). To maintain simplicity, we concentrate on the mathematical structure of the GAP demonstrated in Equations (1), (2), (3b) and (4).

The features of the gymnastics personnel scheduling model include shift scheduling (or hours of the day) and days off scheduling (or days of the week) (Baker, 1976; Bechtold et al., 1991; Alfares, 2004; Brucker et al., 2011; Van den Bergh et al., 2013). In 1977, Henderson and Berry applied a branch and bound algorithm using an integer linear programming model to develop a personnel schedule for telephone operators for 15 minute shifts in a 24 hour timeframe

(Henderson & Berry, 1977). The employees were assumed to be available for all shifts, ignoring variability of employee availability or employee job qualification.

In 1990, Thompson compared two linear programming models in regard to shift scheduling where the staffing pool is only available for particular shifts and where employees are qualified to perform all job tasks (Thompson, 1990). Thompson concentrated specifically on the management of services versus the management of production. We assume all employees are available for all shifts but qualified to perform one, some or all job tasks.

A key contribution of the gymnastics model is the utilization of cross-trained employees and the impact on staffing costs. We use a mixed, heterogeneous workforce where full time and part time employees are qualified to perform multiple job tasks within a facility but not necessarily all job tasks (Alfares, 2004).

Mahhotra and Ritzman utilized a simulation model to assist in the development of a personnel scheduling model concentrating exclusively on part time, cross-trained employees (Malhotra & Ritzman, 1994). They concluded that the benefit of part time, cross-trained employees is dependent on different operational factors; although, the overall impact of utilizing part time, cross-trained employees reduced staffing cost. We expand the scope of our model to part time and full-time cross-trained employees.

Brusco and Johns developed a personnel scheduling model using an integer linear program to minimize staffing costs and incorporating cross-trained employees within a single work shift (Brusco & Johns, 1998). They concluded that having a cross-trained workforce allows for better scheduling flexibility; although, they did not look at job qualification of each employee. We will expand the problem to a 7-day planning period using multiple shifts in a 12-hour workday with employee availability parameters.

In 2002, Gomar, Haas and Morton evaluated the effect of using a cross-trained workforce in the construction industry where multiple projects with a variety of skills need to be completed (Gomar et al., 2002). The objective minimized the employee turnover costs (hiring and firing) and found that cross-trained employees were always preferred. Our objective minimizes total staffing cost related to the payrate of the employee and the hours worked by that employee.

Later, Say and Karabat apply a two stage optimization model to evaluate the trade-off between the output of the company and the skill improvement of the employee when using cross-trained employees (Sayın & Karabatı, 2007). Say and Karabat concentrate primarily on employee skill improvement. We utilize the flexibility of a cross-trained staffing pool but concentrate on the staffing cost and the path to optimality of coach-to-program reassignments.

More recently, Rahimian, Akartunali and Levine implement a hybrid approach using integer programming and variable neighborhood search algorithms to address the nurse rostering problem (Rahimian et al.). The nurse rostering problem has similar features as the gymnastics scheduling problem including a restricted number of shifts scheduled in a day, employee availability constraints and a limited total number of workdays in the planning period. We incorporate different job tasks and allow the employee to change tasks multiple times within the scheduled work day. In addition, we will address our personnel scheduling problem using an integer programming model to accommodate the level of complexity in the structure of the problem.

2.2 Optimality Assessment

At times, knowing what is optimal is not sufficient and decision makers are interested in exploring suboptimal solutions due to other issues associated with application, implementation, and unarticulated objectives. Most assessments of suboptimal versus optimal solutions are focused on a cost or utility, not the actual number of reassignments. We focus on cost and the number of reassignments.

In 1968, Murty propose a partitioning technique to rank the *k* best assignments in order of increasing cost (Murty, 1968). In 2008, small modifications of Murty's technique led to the development of an algorithm with a sub-optimal starting point and a search method for the best improvement rather than the lowest degradation (Pedersen et al., 2008). However, that gives the *k*-best solutions, but does not directly provide information on the number of reassignments required.

In 2004, Zulch, Rottinger, and Vollstedt acknowledge the issue of reassigning personnel to tasks in a manufacturing environment (Zülch et al., 2004). In an attempt to improve overall utility which includes average lead time, system output, average workload, and labor costs, personnel are reassigned; however, the individual cost of reassignment is not evaluated.

Today, optimality assessment is popular in the computer sciences (Casavant & Kuhl, 1988; Raza et al., 2011; Kim, 2013). Indeed, computer resources need to be assigned for tasks and reassigned as the priorities of tasks or parameters change with the goal of conserving computing resources. In Gounaris, Yfoulis, and Paton (2012), the purpose is not only to assign tasks to resources but to reassign tasks only when it is worth the disturbance. (Gounaris et al., 2012). Again, it considers a cost of moving from the current state to an optimum state but not the number of reassignments that are required or desired. This can be challenging if the cost or utility is difficult to measure; therefore, we propose a multiple objective optimization technique to determine the *n* best workload reassignments.

3.0 Mathematical Model

The resulting unique combination of personnel scheduling features creates complexity in daily gymnastic facility operations presented in this model. We address the problem of scheduling a sufficient amount of coaches for each shift for each day along with the time-on-task assignment over a 7-day operation while ensuring all employees scheduled are qualified to perform that task. Gymnastics facilities need a weekly personnel schedule that identifies the most cost-effective set of assignments using a cross-trained workforce with varying availability. We introduce an integer programming model that incorporates staffing cost, employee availability, a cross-trained workforce in order to identify the optimal (utopian) set of coach-to-program assignments.

Unfortunately, the optimal set of coach-to-program assignments differs considerably from the status quo, requiring multiple reassignments from the current state to reach optimality. Making these adjustments with minimal coach reassignment is ideal in helping the children feel at ease while maintaining staffing cost. We then identify the *n* best coach-to-program reassignments from the current state that provides the greatest progress toward the optimal staffing cost solution. The decision maker of the gymnastics facility can visualize the trade-off between the number of coach-to-program reassignments and the resulting progress achieved toward the optimal solution to make a personnel scheduling decision.

3.1 A Model for Weekly Personnel Scheduling for a Gymnastics Facility

The features of this personnel scheduling model include task assignment, shift scheduling and days off scheduling for cross-trained employees with varying availability for a 7-day planning period to be used for an 8-week session. Our objective is to create a coach-to-program assignment plan while minimizing staffing cost. Staffing cost is calculated using the payrate of each employee (c_i) provided by the business owner and the total number of scheduled hours for that employee.

The gymnastics facility considered in this research is located in Raleigh, NC and is one of the largest in the United States. It operates 7 days a week with forty-eight 15-minute periods a day. Employees must be scheduled between 9am and 9pm. If an employee is scheduled to work on a particular day, a minimum of 2 hours and a maximum of 8 hours must be scheduled. If an employee is assigned to teach a specific class, the employee must complete the class before being assigned another class. Employees can work split shifts and work days do not need to be consecutive; however, the number or work days cannot exceed 5 over the 7-day weekly operation.

Demand levels are based on the production by time period and converted to integer values that identifies the number of coaches needed for job task (or program) j during period k on day l, denoted by r_{jkl}. Staffing requirements based on production levels are provided by the business owner and adhere to the coach-to-gymnast ratio expectations. See Table 3-1 for a list of job tasks, program titles and coach-to-gymnast ratios.

Table 3-1 : Gymnastic Facility Job Tasks and Coach-to-Gymnast Ratio

Job Task	Program Title	Coach-to-Gymnast Ratio
1	Parent and Toddler	1:10
2	Preschool	1:7
3	Advanced Preschool	1:7
4	Beginner Girls	1:7
5	Beginner Girls (1.5)	1:7
6	Beginner Boys/Ninja	1:7
7	Intermediate Girls	1:8
8	Intermediate Boys	1:8
9	Advanced Girls	1:8
10	Advanced Boys	1:8
11	Tumble/Advanced Tumble	1:10
12	PreTeam 1	1:12
13	PreTeam 2	1:12
14	Team 1	1:12
15	Team 2	1:12
16	Floater	N/A

A key contribution of this model is the utilization of cross-trained employees with varying availability and the impact on staffing cost. We use a mixed, heterogeneous workforce where full-time and part time employees are qualified to perform multiple job tasks within a facility but not necessarily all job tasks (Alfares, 2004).

Employee availability is provided by the business owner. The business owner guarantees 70% of requested employee availability hours, where a 25-hour part time employee can be scheduled for 17.5 to 25 hours per week. The minimum number of available hours is denoted by w_i and the maximum number of available hours is denoted by v_i. We do not allow overtime in our model.

As noted in Table 3-1, there are 16 job tasks and employees can be qualified to perform one, some or all of these job tasks. Job task assignments for an employee can only change when the class or program changes. Parameter s_{ij} is equal to 1 if employee *i* is qualified to perform task *j*; 0 otherwise.

Let the binary variable x_{ijkl} equal 1 if employee *i* is scheduled to work task *j* in period *k* on day *l*; and 0 otherwise. Let the binary variable y_{il} equal 1 if employee *i* is scheduled to work on day *l*; and 0 otherwise. Let l_{jk} equal *q* if job task *j* starts in period *k* and lasts for *q* additional periods; and 0 otherwise. An integer linear programming model for this problem is given below in equations (5) – (15).

$$MIN\ \sum_{j=1}^{16}\sum_{k=1}^{48}\sum_{l=1}^{7} c_i x_{ijkl}, \forall i \tag{5}$$

Subject To:

$$\sum_{j=1}^{16}\sum_{k=1}^{48}\sum_{l=1}^{7} x_{ijkl} \geq 4v_i\,, \forall i \tag{6}$$

$$\sum_{j=1}^{16}\sum_{k=1}^{48}\sum_{l=1}^{7} x_{ijkl} \leq 4w_i\,, \forall i \tag{7}$$

$$\sum_{k=1}^{48}\sum_{l=1}^{7} x_{ijkl} \leq 350(s_{ij}), \forall i, j \tag{8}$$

$$\sum_{j=1}^{16} x_{ijkl} \leq 1, \forall i, k, l \tag{9}$$

$$\sum_{l=1}^{7} y_{il} \leq 5, \forall i \tag{10}$$

$$\sum_{j=1}^{16}\sum_{k=1}^{48} x_{ijkl} \geq 8(y_{il}), \forall i, l \tag{11}$$

$$\sum_{j=1}^{16}\sum_{k=1}^{48} x_{ijkl} \leq 32(y_{il}), \forall i, l \tag{12}$$

$$\sum_{i=1}^{n} x_{ijkl} \geq r_{jkl}, \forall j, k, l \tag{13}$$

$$\mathrm{l}_{jk}\mathrm{x}_{ijkl} - \sum_{\mathrm{m}=k+1}^{k+\mathrm{l}_{jk}} x_{ij\mathrm{m}l} \leq 0,\ \forall i, j, k \text{ and } l \tag{14}$$

$x_{ijkl}, y_{il} \in \{0,1\}, \forall i, j, k, l$ (15)

Equation (5) is the objective to minimize staffing cost using the payrate of each employee (c_i) and the total number of scheduled hours. Equations (6) and (7) require each employee be scheduled the total number of hours available based on full-time or part time status. Equation (8) ensures each employee is qualified to work job task j if the employee is assigned to work that day.

Equation (9) ensures every employee is assigned only one job task per period if the employee is assigned to work that day. Equation (10) allows employees to only be scheduled for a maximum of 5 days in a 7-day week. Equations (11) and (12) require the employee to work 8 to 32 periods (or 2 to 8 hours) a day if the employee is assigned to work that day.

Equation (13) requires the total number of employees scheduled to meet the staffing requirements for each task for each period for each day based on production schedules. If an employee is assigned to teach a specific class, equation (14) ensures the employee teaches completes the class before being assigned another class. Equation (15) requires the assignment decision variables x_{ijkl} and y_{il} to be binary.

3.2 Modifying the Model for Tradeoffs

The solution to the integer linear programming model in (5) - (15) provides an optimal set of coach-to-program assignments in a significantly less amount of time than the manual process. However, the optimal solution requires a large number of program reassignments from the current state, which disrupts established coach-to-gymnast relationships. In other words, the mathematical model in (5) - (15) is indifferent between an assignment plan that achieves the utopian solution's optimal objective in 200 program reassignments and one that achieves the same utopian objective function value in 50 reassignments. This is a vital consideration as nervous and anxious children may not want to come to class or continue with gymnastics if the familiar coaching face has been changed.

It is reasonable to assume that the business owner would likely prefer to make 30 reassignments rather than 200. Thus, a solution involving 30 reassignments that achieves 98% of the utopian objective function value might actually be preferable to the business owner. Using an extension of the integer linear programming model in equations (5) – (15), we add equation (16) where x^*_{ijkl} denotes the set of actual coach-to-program assignments. Let x^*_{ijkl} equal one for all current state variables; 0 otherwise. Let λ denote an integer value representing the number of allowable reassignments.

$$\sum x^*_{ijkl} = \lambda \ \forall\, i, j \in \Omega, \text{ where } \lambda \text{ is Integer and } \Omega = \left\{i, j, k, l \;\middle|\; x^*_{ijkl}\right\} \tag{16}$$

Equation (16) requires the integer linear programming model to use a set number (λ) of reassignments. The integer value, λ, is calculated by identifying the sum of current state variables and subtracting the number of desired reassignments. This enables the decision maker to assess the number of reassignments in comparison to the resulting progress to the utopian solution.

4.0 Application

We demonstrate the value of our multi-objective method for evaluating the trade-off between the number of program reassignments and the resulting progress achieved toward the optimal staffing cost in three scenarios. As an added benefit, computerizing the scheduling tool is critical in the day-to-day life of a gymnastics business owner allowing the ability to adjust and measure the impact on staffing cost in a timely manner.

4.1 Identification of Optimal Solution

We utilized data from one of the largest gymnastics facilities in the U.S. In 2016, this facility was the voted the top gymnastics facility in Raleigh, NC with an enrollment exceeding 1,500 gymnasts. We consider two 8-week sessions to demonstrate the trade-off assessment methodology. Table 3-2 shows the breakdown of each session with regard to the session dates, number of employees, number of programs and the number of 15-minute periods. A total of 16 job tasks (or programs) are offered and a typical program schedule is provided in Tables 3-3 and 3-4.

Table 3-2 : Gymnastics Optimality Trade-Off Sessions

Session	Session Length	# of Employees	# of Programs	# of 15-Minute Periods
5	April 10-June 4	39	278	1598
6	June 5-July 30	43	289	1679

Table 3-3 : Example of Weekly Gymnastics Class Schedule for Lower Level Programs

Day	Parent & Toddler	Preschool	Advanced Preschool	Beginner Girls	Beginner Girls (1.5)	Beginner Boys/Ninja	Intermediate Girls	Intermediate Boys
Monday	9:30-10:15 4:00-4:45 6:30-7:15	2:00-3:00 4:00-5:00 6:00-7:00	10:30-11:30 2:00-3:00 5:00-6:00 6:00-7:00 7:00-8:00	4:00-5:00 5:00-6:00 6:00-7:00 7:00-8:00	5:30-7:00	4:00-5:00	4:00-5:30 5:30-7:00 7:00-8:30	4:00-5:30
Tuesday	9:30-10:15 10:30-11:15 6:30-7:15	9:30-10:30 10:30-11:30 5:00-6:00 7:00-8:00	9:30-10:30 10:30-11:30 2:00-3:00 4:00-5:00 6:00-7:00 7:00-8:00	2:00-3:00 4:00-5:00 5:00-6:00 6:00-7:00 7:00-8:00	4:00-5:30	4:00-5:00 7:00-8:00	4:00-5:30 5:30-7:00 7:00-8:30	5:30-7:00
Wednesday	9:30-10:15 10:30-11:15 4:00-4:45 6:30-7:15	9:30-10:30 2:00-3:00 4:00-5:00 5:00-6:00 6:00-7:00	10:30-11:30 1:00-2:00 2:00-3:00 4:00-5:00 5:00-6:00 6:00-7:00	3:00-4:00 4:00-5:00 5:00-6:00 6:00-7:00 7:00-8:00	5:30-7:00	5:00-6:00	4:00-5:30 5:30-7:00 7:00-8:30	4:00-5:30
Thursday	9:30-10:15 10:30-11:15 6:30-7:15	9:30-10:30 10:30-11:30 4:00-5:00 6:00-7:00 7:00-8:00	2:00-3:00 4:00-5:00 5:00-6:00	4:00-5:00 5:00-6:00 6:00-7:00 7:00-8:00	None	4:00-5:00 5:00-6:00 6:00-7:00	4:00-5:30 5:30-7:00 7:00-8:30	4:00-5:30 5:30-7:00
Friday	None	None	9:30-10:30	None	None	None	4:00-5:30 5:30-7:00	None
Saturday	9:00-9:45 10:00-10:45 11:00-11:45	9:00-10:00 10:00-11:00 11:00-12:00	9:00-10:00 10:00-11:00 11:00-12:00	9:00-10:00 10:00-11:00 11:00-12:00	10:30-12:00 12:00-1:00	10:00-11:00	9:00-10:30 10:30-12:00 12:00-1:30	11:00-12:30
Sunday	None	1:00-2:00 2:00-3:00	1:00-2:00 2:00-3:00 3:00-4:00	1:00-2:00 2:00-3:00 3:00-4:00	None	1:00-2:00 3:00-4:00	2:30-4:00 3:00-4:30	None

Table 3-4 : Example of Weekly Gymnastics Class Schedule for Upper Level Programs and Floater

Day	Advanced Girls	Advanced Boys	Tumble/ Advanced Tumble	PreTeam 1	PreTeam 2	Team 1	Team 2	Floater
Monday	4:00-6:00 5:00-7:00 6:00-8:00	None	7:15-8:15 8:00-9:00	4:00-5:15 5:15-7:15	None	4:15-6:30 6:15-8:15	4:15-7:15	4:00-5:00 5:00-6:00 6:00-7:00 7:00-8:00
Tuesday	4:00-6:00 5:00-7:00 6:00-8:00	None	7:15-8:15 8:00-9:00	4:15-6:15 6:15-8:45	None	4:15-6:15 5:45-8:15	5:45-8:45	4:00-5:00 5:00-6:00 6:00-7:00 7:00-8:00
Wednesday	None	None	7:15-8:15	4:15-5:45 5:45-7:15	None	4:15-6:30 6:15-8:15	4:15-7:15	4:00-5:00 5:00-6:00 6:00-7:00 7:00-8:00
Thursday	4:00-6:00 6:00-8:00	6:00-8:00	7:15-8:15 8:00-9:00	4:15-6:15 6:15-8:45	None	4:15-6:15 5:45-8:15	5:45-8:45	4:00-5:00 5:00-6:00 6:00-7:00 7:00-8:00
Friday	None	None	None	4:00-5:15 4:15-5:45 5:30-7:00	5:30-7:00	4:15-6:15 6:15-8:15	5:45-8:45	4:00-5:15
Saturday	9:00-11:00 11:00-1:00 12:00-2:00	None	12:30-1:30	None	12:00-2:00 2:00-4:00	9:00-11:15 10:45-1:30	1:00-5:00	9:00-10:00 10:00-11:00 11:00-12:00 12:00-1:00
Sunday	2:00-4:00	None	None	None	None	None	None	1:00-2:00 2:00-3:00 3:00-4:00

The integer linear programming model for this problem was optimized using IBM ILOG CPLEX Optimizer 12.7. Table 3-5 shows the mathematical model characteristics by session, specifically the number of binary variables and constraints.

Table 3-5 : Gymnastics Optimality Model Characteristics by Session

Session	# of Binary Variables	# of Constraints
5	209,937	439,368
6	231,469	483,880

Table 3-6 shows the optimization results for the utopian objective value for each session. Each session's objective value is calculated within 0.5 percent of optimality or 2 hours of CPU time. The utopian objective value minimizes the total staffing cost while utilizing a cross-trained staffing pool. In session 5, the utopian objective value is less than the actual objective value by 744 dollars, while using an additional 151 15-minute time period assignments. In session 6, the utopian objective value is less than the actual objective value by 2,833 dollars, while using an additional 76 15-minute time period assignments. We are able to provide an optimal solution that efficiently staffs the programs while reducing cost.

Table 3-6 : Gymnastics Optimality Model Objective Value Results versus Actual

Session	Percent to Optimality	CPU Time (seconds)	Utopian Staffing Cost ($)	Utopian # of 15-Minute Periods	Actual Staffing Cost ($)	Actual # of 15-Minute Periods
5	100%	5202	28,406	1745	29,150	1594
6	100%	5617	28,716	1813	31,549	1737

4.2 Closeness to Optimality Assessment

The current set of coach-to-program assignments was provided by the business owner for each session using the program registration data. The established current state variables (x_{ij}^*)

were set equal to 1. Using the integer linear programming model for evaluating the closeness to optimality based on number of reassignments, the tradeoff assessment is identified.

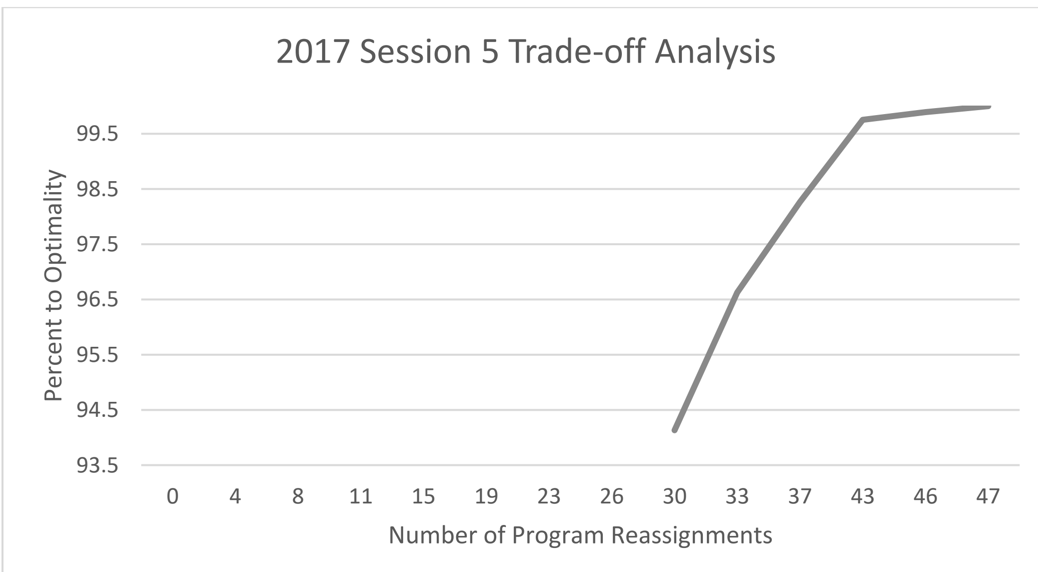
We optimized each session for various levels of λ. In other words, we allowed 0 to 50 program reassignments (equivalent to 0 to 350 15-minute period reassignments) to be made from the current state. Each value of λ requires the model to be optimized separately. Here, each scenario's tradeoff assessment is calculated within 0.5 percent of optimality or 1 hour (3600 seconds) of CPU time.

The utopian solution for Session 5 requires 259 program reassignments of a total of 278 programs and operates at 28,406 dollars a week. After implementing the personnel scheduling multiple objective optimization technique, the model returned infeasible solutions for 0 to 29 allowable program reassignments from the current state. This demonstrates that the current state did not meet the constraints required to perform adequately given the resources. When we allowed 30 program reassignments, we reached 94.13% to utopian solution with an objective value of $30,073. We reach 100% of optimality with an objective value of $28,406 in 1552 seconds of CPU time when we allowed 47 program reassignments. See Table 3-7 and Figure 3-1 for details.

Table 3-7 : Session 5 Results Summary for Trade-off Assessment by Scenario

	# of Program Reassignments to Feasible Solution	Objective Value	% toward Utopian Solution	CPU Time (seconds)
Utopian	**259**	**28,406**	**100%**	**5202**
	0 to 29	Infeasible	Infeasible	Infeasible
	30	$30,073	94.1%	2159
	33	$29,364	96.6%	3600
	37	$28,899	98.3%	2713
	43	$28,476	99.8%	2893
	46	$28,437	99.9%	3600
	47	$28,406	100%	1552

Figure 3-1 : Session 5 Trade-off Assessment: Closeness to Optimality versus Number of Reassignments

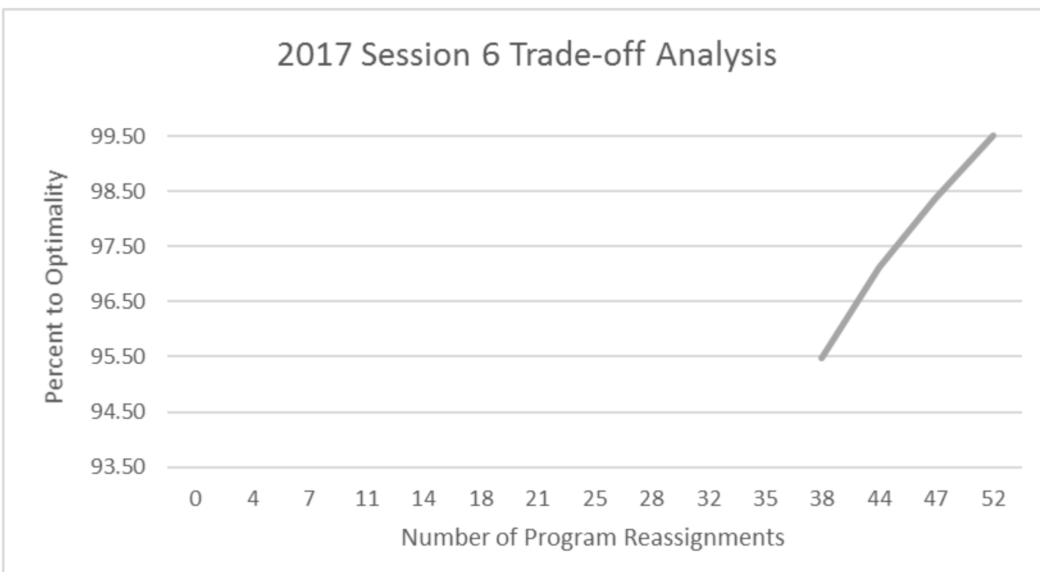


The utopian solution for Session 6 requires 272 program reassignments of a total of 289 programs and operates at $28,716 a week. After implementing the personnel scheduling multiple objective optimization technique, the model returned infeasible solutions for 0 to 37 allowable program reassignments from the current state. Again, this demonstrates that the current state did not meet the constraints required to perform adequately given the resources. When we allowed 38 program reassignments to be made, we reached 95.5% to utopian solution with an objective value of $30,013. We reach 99.5% of optimality with an objective value of $28,854 in 3600 seconds of CPU time when we allowed 50 program reassignments. See Table 3-8 and Figure 3-2 for details.

Table 3-8 : Session 6 Results Summary for Trade-off Assessment by Scenario

	# of Program Reassignments to Feasible Solution	Objective Value	% toward Optimality	CPU Time (seconds)
Utopian	**272**	**28,716**	**100%**	**5617**
	0 to 37	Infeasible	Infeasible	Infeasible
	38	$30,013	95.5%	3600
	44	$29,545	97.11%	3295
	47	$29,184	98.4%	1874
	50	$28,854	99.5%	3600

Figure 3-2 : Session 6 Trade-off Assessment: Closeness to Optimality versus Number of Reassignments



5.0 Conclusion

This research addresses the coach-to-program task assignment and tour scheduling problem in gymnastics facilities. Typically, gymnastic facility business owners require an extensive amount of time and frustration to complete the formulation of an 8-week personnel schedule. This is not easily adjusted due to the complexity of hard copy calculations. We identify the optimal staffing cost assignment plan using cross-trained employees and varying employee availability using an integer linear programming problem.

The optimal assignment requires multiple program reassignments from the current state to reach optimality; therefore, we propose a technique to identify the *n* best coach-to-program reassignments from the current state that provide the greatest progress toward the optimal staffing cost solution. The decision maker can visualize the trade-off between the number of program reassignments and the resulting progress achieved toward the optimal staffing cost. The decision maker can make an informed personnel scheduling decision that provides the level of disruption or reassignment and staffing cost desired.

Chapter 4 : Optimizing Product Test Scheduling with In-Process and At-Completion Inspection Constraints

Optimizing Product Test Scheduling with In-Process and At-Completion Inspection Constraints

ABSTRACT

This research focuses on product test scheduling in the presence of in-process and at-completion inspection constraints. Such testing arises in the context of the manufacture of products that must perform reliably in extreme environmental conditions. Each product receives a certification at the successful completion of a predetermined series of tests. Operational efficiency is enhanced by determining the optimal order and start times of tests so as to minimize the make span while ensuring that technicians are available when needed to complete in-process and at-completion inspections We first formulate a mixed-integer programming model (MILP) to identify the optimal solution to this problem using IBM ILOG CPLEX Interactive Optimizer 12.7. We also present a genetic algorithm (GA) solution methodology that is implemented and solved in Microsoft Excel. Computational results are presented demonstrating the relative merits of the MILP and GA solution approaches across a number of scenarios.

1.0 Introduction

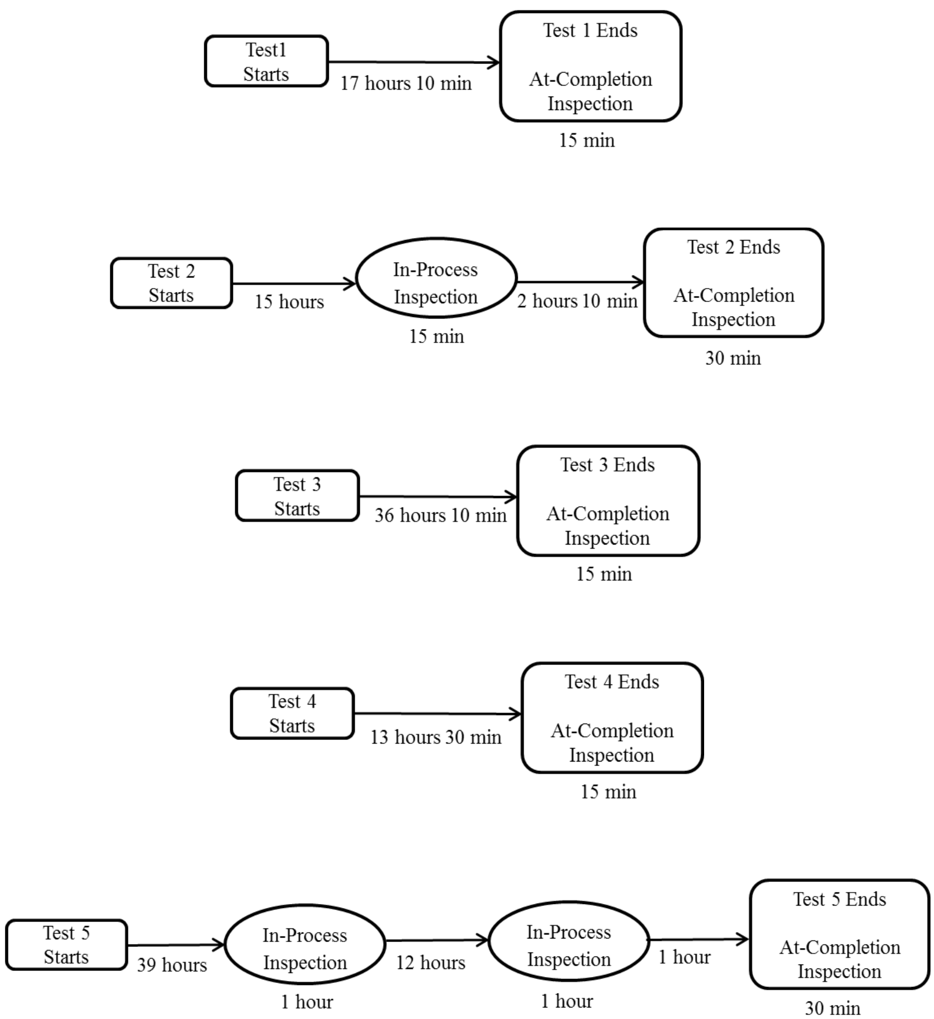
As the recent Deepwater Horizon oil spill illustrates, failure of industrial equipment in sensitive environmental settings can have devastating impacts on the natural world as well as severe economic implications for the associated business entities (Barstow, Rohde, & Saul, 2010). As a result, the scheduling of testing on products under extreme environmental conditions presents both a complex and critically important business problem. Various products need to be exposed to low and high temperatures, different vibration levels, and variations in pressure and humidity in isolated chambers as part of the market certification process. Thus, the certification process might involve a series of tests that each require in-process and at-completion inspections by personnel to validate the functionality and durability of the product before delivery and deployment.

In order to operate effectively, companies must be able to minimize lead times, provide accurate delivery dates, and deploy resources in an effective manner. In the context of this research, all these things necessitate an ability to identify the optimal order and start times of products tests in such a way as to minimize the make span while ensuring that the appropriate technicians are scheduled to be available to complete the required in-process and at-completion inspections as needed.

The problem considered here was brought to our attention by engineers at Rockwell Automation, the world's largest company dedicated to industrial automation, with 2016 revenue of $5.8 billion. For the product in question, a series of five required environmental condition tests must be carried out on each unit in order for it to receive certification. These tests are summarized in Figure 4-1. We refer to the tests as Test 1, Test 2, Test 3, Test 4, and Test 5;

however, the tests are independent and may be completed in any order. All five tests require an inspection at the end of the test, referred to as an “at-completion” inspection. Additionally, Tests 2 and 5 require inspections at different points during while the tests are being carried out, referred to as “in-process” inspections. An employee must be present for every inspection. However, employees do not need to be present to start a test as the tests can be scheduled to start automatically at any time on any day. Our objective is to determine a schedule (testing order and start times) that minimizes the make span of the 5 tests while ensuring all required inspections take place during normal operating hours (or as close to that as possible).

Figure 4-1 : Summary of Required Tests



We first identify a mixed-integer programming model (MILP) to identify the optimal order of tests that minimizes the completion time using a specialized software, IBM ILOG CPLEX Interactive Optimizer 12.7. We then identify a spreadsheet implementation of this problem that can be optimized using the genetic algorithm (GA) in the Solver that comes with Microsoft Excel 2016.

The remainder of this paper is organized as follows. In section 2, a literature review evaluating vehicle routing, product test scheduling and job shop genetic algorithms is provided. Section 3 examines our methodology and techniques used to develop the MILP model and the GA. Section 4 provides a demonstration of the mathematical models and the findings from that implementation followed by conclusions and areas for future work in section 5.

2.0 Literature Review

The nature of the work in the present paper leads to an evaluation of literature in vehicle routing, product test scheduling and job shop genetic algorithms to address variants of these problems. A brief overview of these research fields is provided to support the methodology and techniques used to develop the MILP model and the GA.

2.1 Vehicle Routing

The vehicle routing problem (VRP) was first introduced in 1959 by Dantzig and Ramser researching the optimal route for a fleet of gasoline delivery trucks with multiple gas stations and one central depot (Dantzig & Ramser, 1959). Fifty-five years later the vehicle routing problem and variations of this problem are still of high interest due to its practical relevance.

A standard mathematical formulation of the integer programming model of a directed capacitated vehicle routing problem is provided in equations (1) – (6) (Laporte et al., 1986; Toth

& Vigo, 2001; Golden et al., 2008; Toth & Vigo, 2014). Let the binary variable $x_{ij} = 1$ if a vehicle travels between *i* and *j*; 0 otherwise. The cost incurred if a vehicle travels between *i* and *j* is denoted by c_{ij}. Equation (1) is the objective to minimize the total routing costs. More specifically, the optimal solution identifies which vehicles travel which routes in the least cost manner. We do not use a cost parameter in the test scheduling problem; we concentrate on the total time to complete the five extreme environmental conditions tests.

$$MIN \ \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \tag{1}$$

Subject To:

$$\sum_{j=1}^{m} x_{ij} = 1, \forall i \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \forall j \tag{3}$$

$$\sum_{j=1}^{m} x_{0j} = |K| \tag{4}$$

$$\sum_{j=1}^{m} x_{ij} \geq r(S), \forall \ S \ subset \ of \ N \ where \ S \ cannot \ include \ the \ depot \tag{5}$$

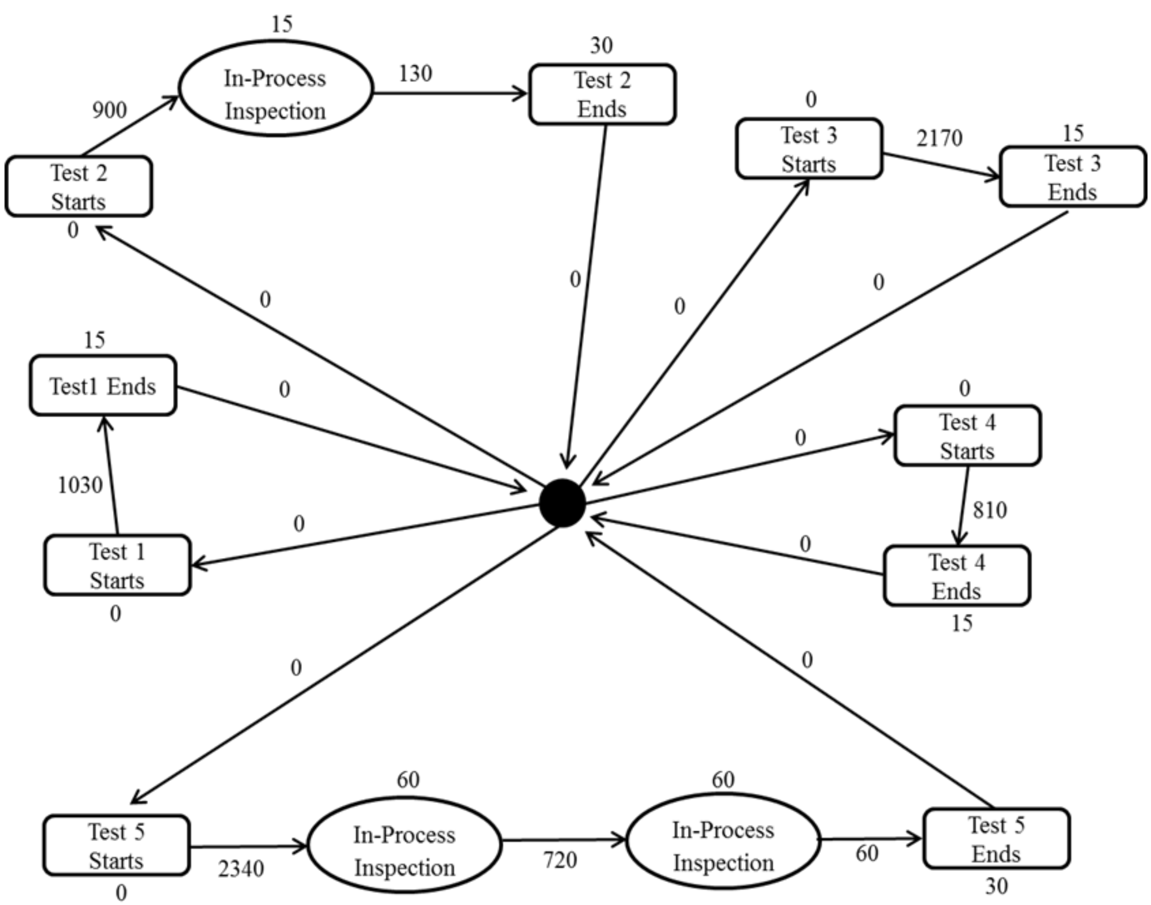
$$x_{ij} \in \{0,1\}, \forall i, j \tag{6}$$

Equations (2) and (3) work together to ensure the travel between *i* and *j* occurs only once. Equation (4) ensures that exactly $|K|$ routes are used from the depot, typically identifying the number of trucks available. Let *N* be the total number of stations that need to be serviced or, in more general terms, the total number of nodes that need to be visited. Equation (5) is a generalized constraint for subtours. We do not use this constraint in the test scheduling problem. Equation (6) requires the decision variable to be binary.

The fundamentals of our test scheduling problem resemble a VRP. The test scheduling problem has five tests that each have established routes that must be traveled in that order by the same vehicle (or in our case, product); the tests can be completed in any order. See Figure 4-2 for a summary representation of the test scheduling problem in a VRP format.

Typically, the VRP objective is to minimize cost (or distance) with a variant of constraints where we want to minimize the total time it takes to travel all five routes. All edges and nodes are given a cost measured in minutes demonstrating the time required to complete that process. Every in-process inspection node and test end node require a technician to be present for that duration of time. The test start nodes and all edges do not require a technician to be present.

Figure 4-2 : VRP Representation of the Test Scheduling Problem



The basic VRP mathematical model is not ideal for a test scheduling problem due to the addition of varying time specific in-process and at-completion inspections by personnel. Consequently, most VRP variants include parameters of travel time with time window constraints (VRPTW) (Toth & Vigo, 2001; 2014). This idea originates from the public transportation process of assigning different types of vehicles to multiple depots with timetabled trips that typically include fixed arrival and departure times at a minimal cost (Bunte & Kliewer, 2009).

Typically, equations (7) and (8) are added to the VRP integer programming model to accommodate the time constraints. The parameter, t_{ij}, identifies the time required to travel from node *i* to node *j* and time windows for each node $[a_i, b_i]$ are added. The variable T_i denotes the start time of service or visit for node *i*. The addition of subscript *k* can accommodate for multiple trucks; although, we do not include this subscript due to the fact we only have one truck (or product) that needs to travel in the system.

$$a_i \leq T_i \leq b_i, \forall i \qquad (7)$$

$$T_i - T_j + M(x_{ij}) \leq M - t_{ij}, \forall i, j \qquad (8)$$

Equation (7) ensures the start time of service occurs in the given time window for node *i*. If the truck travels from node *i* to node *j*, equation (8) ensures that node *j* service starts after node *i* service is complete and the truck has traveled from node *i* to node *j* by using an arbitrarily large positive integer, *M*.

The VRPTW mathematical structure is similar to our test scheduling problem; although, the order of the nodes is dependent on the fixed arrival and departure times of each node. In the test scheduling problem, identifying the start time to conduct each test is necessary but not dependent on a time window. We utilize a variation of equations (7) and (8) to accommodate our time requirements.

Additionally, technician involvement at time specific in-process and at-completion inspections creates a unique scheduling problem. The identification of each time period that requires technician observation is needed to create the personnel schedule, certify the product and meet business operating conditions. The capability of identifying the minimal make span of the product increases the number of products processed and certified.

2.2 Product Test Scheduling

With the importance of safety and the need to reduce production cost, the automotive industry has been the focus of product test scheduling research (Chelst et al., 2001; Bartels & Zimmermann, 2009; Reich et al., 2016; Shi et al., 2017). In 2009, Bartels and Zimmerman evaluate the development cost on automotive R&D projects (Bartels & Zimmermann, 2009). More specifically, the objective is to minimize the number of experimental vehicles by optimizing the test schedule using a mixed-integer linear program formulation. There are a multitude of rules that need to be followed; for example, a subset of tests cannot be completed simultaneously or on the same vehicle. The MILP is industry specific and almost impossible to use in other industries.

Most recently, Reich et al. use an integer program in combination with a column-generation algorithm to generate an automobile crash-test schedule for Ford (Reich et al., 2016). The primary objective is to maximize the utilization of vehicles for testing to minimize total cost due to the expensive nature of experimental vehicles. Each optimization evaluates a different sequence of tests and the impact on cost. In our paper, we minimize time needed to complete all tests on one product where total time is a parameter in the Reich et al. configuration.

2.3 Job Shop Genetic Algorithms

The job shop scheduling problem (JSP) has many similar characteristics as the test scheduling problem. The JSP is the process of assigning *n* different jobs to *m* different machines. The test scheduling problem is the process of assigning *n* different tests to specific time periods to ensure technician involvement during business operating hours. The number of machines required is not a factor in our problem. With that said, the number of similarities is too strong to neglect this area of research.

In the JSP, each job requires a sequence of operations and each operation must be completed for that job to be considered complete. Jobs are independent of each other and the processing times are fixed. The transportation time between jobs is included in the processing times. Only one job can be processed at a time on a machine. The operations and jobs can be processed in any order. In our test scheduling problem, the tests are jobs and the different testing time and inspections are operations; although, in the test scheduling problem the operations must be completed in order and at specific time periods. Identifying the optimal order and set of assignments while minimizing the make span is almost impossible due to the NP-hard nature of the combinatorial optimization JSP; therefore, genetic algorithms are used to solve these problems to optimality (Garey et al., 1976; Cheng et al., 1996; 1999; Potts & Strusevich, 2009).

Cheng, Gen and Tsujimura identified a list of nine categories for genetic algorithms used in attempt to solve the JSP; unfortunately, all categories utilize multiple machines and the basic assignment problem of job order does not factor in technician involvement. One category resembled the test scheduling problem called priority rule-based representation. Job order is decided based on priority dispatch rules. Dispatch rules help decide which operation is next in line for completion. Rules can incorporate the shortest processing time, longest processing time, earliest due date, first come first serve, or even random assignment. To date, business operating hours and technician involvement has not been researched; therefore, the genetic algorithms used do not apply to the test scheduling problem.

3.0 Models for the Test Scheduling Problem

We now describe two different approaches to solving this problem. We first develop a MILP to identify the optimal scheduling of tests that minimizes the completion time to conduct tests for one product unit. Next, we propose a GA-based solution methodology that may be implemented using Microsoft Excel 2016.

3.1 A Mixed-Integer Linear Programming Model

We first address the problem by identifying the mixed-integer programming model. Let N denote the number of tests to be performed. Let the binary variable $x_{ij} = 1$ if test *j* follows test *i*; and 0 otherwise. Let the binary variable $s_{ij} = 1$ if test *i* starts in time period *j*; and 0 otherwise. Let the variable t_i denote the time period in which test *i* begins. Finally, let the variable Q denote the make span to conduct all extreme environmental conditions tests with in-process and at-completion inspections on one product.

Automated equipment allows for testing to begin at any time during a 24-hour time frame. Thus, an important aspect of this problem is the granularity of time period measurement (e.g., seconds, minutes, *n*-minute intervals, hours, etc). For the problem addressed in this research we selected 5 minute intervals across a three-week planning horizon. This results in

6048 unique time periods (denoted as TP) in which tests may start or be carried out. We let the parameter l_i denote the number of time periods required to run test *i* including in-process and at-completion inspection times. The facility normally operates with personnel present Monday through Friday (5-day work week) from 7am to 6pm (11-hour work day). However, the facility allows personnel to arrive one hour early or stay two hours late to accommodate Test 5's in-process and at-completion inspections. Again, employees do not need to be present to start a test as each test can be scheduled to start automatically at any time and the tests can be carried out in any order. We let the parameter b_{ij}= 1 if the employees are available for the required inspections of test *i* during time period *j*; 0 otherwise.

The MILP model for this problem is given below in equations (9) - (22). Our objective is to minimize the completion time to conduct all tests with in-process and at-completion inspections on one product. The objective function in equation (9) and constraint in equation (17) work together to minimize the make span of all testing. More specifically, equation (17) requires that all elapsed time periods between the start and end of all testing must be less than Q.

$$MIN\ \mathrm{Q} \tag{9}$$

Subject to:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} = N - 1\ , \forall i, j\ where\ i \neq j \tag{10}$$

$$\sum_{i=1}^{n} x_{ij} \leq 1, \forall j\ where\ i \neq j \tag{11}$$

$$\sum_{j=1}^{n} x_{ij} \leq 1, \forall i\ where\ i \neq j \tag{12}$$

$$\sum_{j=1}^{\mathrm{TP}-l_i} s_{ij} = 1, \forall i \tag{13}$$

$$t_i - \sum_{j=1}^{\mathrm{TP}-l_i}(j * s_{ij}) = 0, \forall i \quad (14)$$

$$t_i + l_i \leq t_j + M\left(1 - x_{ij}\right), \forall i, j \; where \; i \neq j \quad (15)$$

$$s_{ij} \leq b_{i(j+k)}, \forall i, j, and \; k \in P_i \quad (16)$$

$$t_i + l_i - t_j \leq \mathrm{Q}, \forall i, j \; where \; i \neq j \quad (17)$$

$$t_i \geq o, \forall i \quad (18)$$

$$t_i \leq c \; + M(d_i), \forall i \quad (19)$$

$$\sum_{i=1}^{N} d_i \leq 4 \quad (20)$$

$$s_{ij}, x_{ij}, d_i \in \{0,1\}, \forall i, j \quad (21)$$

$$t_i, Q \;\geq 0 \quad (22)$$

Equation (10) requires the product to switch between tests a total of (*N*-1) times ensuring every test is completed. Equation (11) ensures that the product switches to test *j* only once and equation (12) ensures that the product switches to test *i* only once. Equation (13) requires each test be assigned a starting time period.

Equation (14) requires t_i to equal the time period in which test *i* begins. In other words, it is the link between t_i and s_{ij}. When the product switches from test *i* to test *j*, equation (15) ensures that test *j* starts after test *i* is completed by using an arbitrarily large positive integer, *M*. Equation (16) ensures all in-process and at-completion inspections occur when personnel are available during business operations. When an inspection is required in the k^{th} time period during and/or at the end of test *i*, $k \in P_i$.

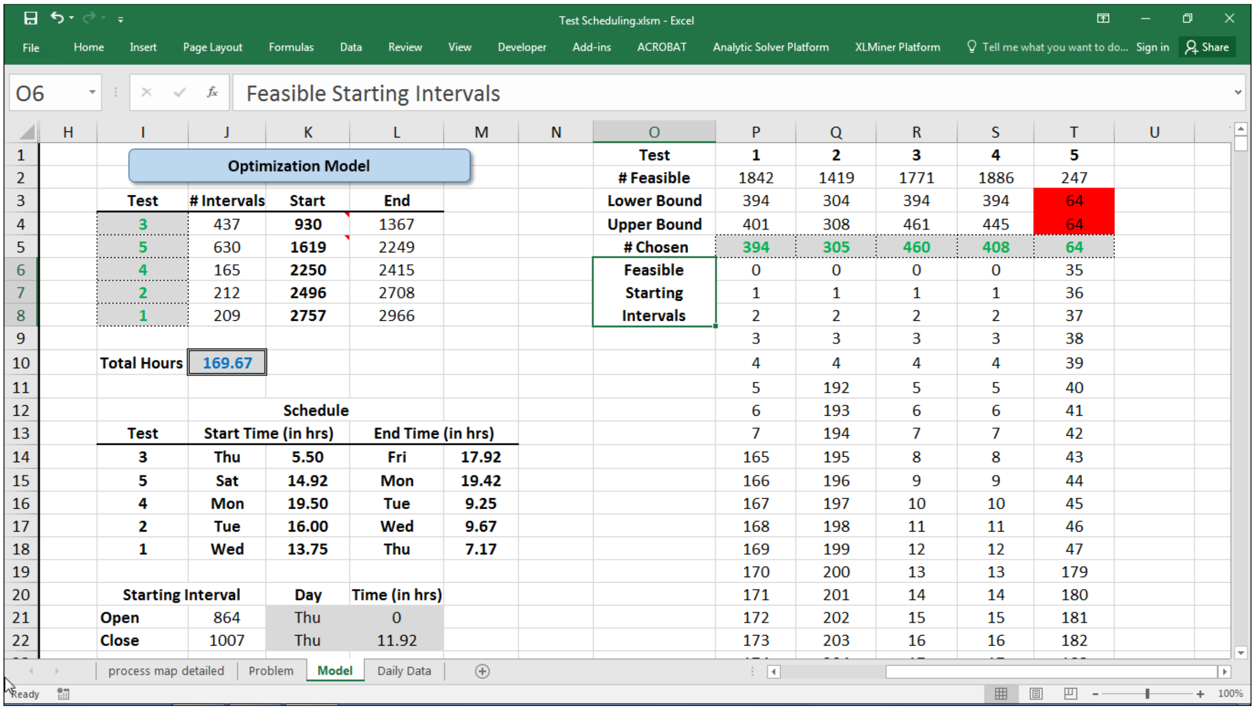
In the event the testing cycle needs to start in a specific time window, for example Tuesday morning, equation (18) will ensure the first test will start at or after a specific time period, o. Additionally, equations (19) and (20) work together to ensure that the first test occurs before the time window closes at a specific time period, c. Equation (21) requires the assignment decision variables x_{ij}, s_{ij} and d_i to be binary. Equation (22) imposes non-negativity constraints for the remaining decision variables, t_i and Q.

3.2 A Genetic Algorithm Model

A careful consideration of equation (16) in the MILP model suggests another approach to solving this problem. Recall that the binary variable $s_{ij} = 1$ if test *i* starts in time period *j*; and 0 otherwise. However, equation (16) eliminates starting periods that would place in-process or at-completion inspections at times when needed workers are not available (e.g., weekends and outside of normal working hours). Before solving the problem, we can use this same idea to create a list of feasible start times for each test. Because we desire to complete the entire suite of tests in the shortest possible amount of time, for a given sequence of tests (e.g., Test 2, Test 3, Test 5, Test 1, Test 4), it follows that at the completion of any test we should begin the next test in the sequence at its next available feasible starting time. Thus, a feasible solution to the problem is given by a permutation of the test numbers along with a feasible starting time period for the first test (from which feasible starting times for the remaining tests can be derived). We must then simultaneously determine the optimal permutation of the tests and the optimal starting time period for the first test in that permutation. This approach is fairly easy to implement and optimize using the Solver built into Microsoft Excel. Moreover, Excel provides a readily available and familiar to solution platform for this type of problem.

Figure 4-3 shows an example of how the test scheduling problem may be implemented in Excel for a testing sequence that must start on a Thursday morning between 12:00 AM and 11:55 AM (or between the 864th (cell J21) and 1007th (cell J22) five-minute time intervals in the planning horizon). We first pre-process the data to identify feasible starting times for each of the tests. The starting times represent 5 minute intervals with each interval being assigned an index number. These index numbers are listed in columns P through T for tests 1 through 5, respectively. In P5 through T5, Solver selects a cell from the list of index numbers associated with a starting time period. This is done for each test in the event that test becomes the first test in the sequence of tests given in cells I4 through I8. Note that test 3 is the first test in the solution shown in Figure 4-3 and it starts in the 930th time period (cell K4), which is the 460th feasible time period for test 3 (cell R5). The remaining starting periods for tests 5, 4, 2, and 1 (in that order) correspond to the first feasible time periods following the end of the preceding tests. The total time required to execute this testing sequence is computed in cell J10 as 169.97 hours. The Solver setting used for solving the problem are given in Figure 4-4.

Figure 4-3 : Example GA Implementation of Test Scheduling Problem in Excel



Cell	Formula	Copied to
J10	=(L8-K4)*5/60	--
K4	=INDEX($P$6:$T$1891,INDEX($P$5:$T$5,1,I4),I4)	--
K5	=INDEX($P$6:$T$1891,MATCH(L4,OFFSET($P$6,0,I5-1,1886,1),1)+1, I5)	K6:K8
L4	=K4+J4	L5:L8
P2	=COUNT(P6:P6050)	Q2:T2
P3	=IFERROR(MATCH($J$21,P$6:P$6048,0),IFERROR(MATCH($J$21,P$6:P$6048,1)+1,1))	Q3:T3
P4	=MAX(IFERROR(MATCH($J$22,P$6:P$6048,0),IFERROR(MATCH($J$22,P$6:P$6048,1),P2)),P3)	Q4:T4

Figure 4-4 : Solver Settings

Solver Settings:

Objective: J10 (Min)

Variable cells: I4:I8 and P5:T5

Constraints:

- I4:I8 all different
- P5:T5 >= P3:T3
- P5:T5 <= P4:T4
- P5:T5 integer

Solver Options:

- Evolutionary Engine

4.0 Computational Results

We evaluate and compare the MILP and the GA with a series of 15 what-if scenarios. The decision maker is faced with the dilemma of selecting the best optimal sequence and start times of tests and indicating when an employee needs to be present to conduct in-process and at-completion inspections. This decision might need to be made on any day of the week as summarized in scenarios 1 through 14 in Table 4-1. Scenario 15 in Table 4-1 is included as a benchmark to identify the best possible time for starting the testing sequence and the best possible outcome that can be expected.

Table 4-1 : Time Windows by Scenario

Scenario	Day	Time Window
1	Monday	12am-11:55am
2	Monday	12pm-11:55pm
3	Tuesday	12am-11:55am
4	Tuesday	12pm-11:55pm
5	Wednesday	12am-11:55am
6	Wednesday	12pm-11:55pm
7	Thursday	12am-11:55am
8	Thursday	12pm-11:55pm
9	Friday	12am-11:55am
10	Friday	12pm-11:55pm
11	Saturday	12am-11:55am
12	Saturday	12pm-11:55pm
13	Sunday	12am-11:55am
14	Sunday	12pm-11:55pm
15	All	Any

Table 4-2 shows the optimal objective value for each scenario for the MILP and the GA. Recall that the objective value is the total time required to complete all 5 tests on one product. The MILP optimal objective value is the same as the GA optimal objective value for all 15 scenarios, demonstrating the adequate performance of the GA. The GA required approximately 5 minutes of CPU time and the MILP solved in seconds. The genetic algorithm performs equally as well in slightly more time using Microsoft Excel 2016.

Scenarios 10 and 11 (Friday PM and Saturday AM) returned infeasible solutions for the MILP and the GA; therefore, the tests cannot be scheduled to start during those times without interruption or waiting for a technician to complete the inspections. Scenarios 5 and 7 return the best make span of the product (or the least amount of time) with an objective value of 169.67 hours.

Table 4-2 : Computational Results by Scenario for the MILP and GA

Scenario	Time Window	MILP Obj. Value (hours)	MILP Run Time (seconds)	GA Obj. Value (hours)	GA Run Time (seconds)
1	Monday AM	171.00	1.72	171.00	319
2	Monday PM	175.33	5.13	175.33	318
3	Tuesday AM	171.00	1.64	171.00	329
4	Tuesday PM	175.33	1.13	175.33	316
5	Wednesday AM	169.67	1.31	169.67	320
6	Wednesday PM	175.33	1.14	175.33	319
7	Thursday AM	169.67	1.11	169.67	339
8	Thursday PM	175.33	0.89	175.33	317
9	Friday AM	171.00	0.75	171.00	326
10	Friday PM	Infeasible	Infeasible	Infeasible	Infeasible
11	Saturday AM	Infeasible	Infeasible	Infeasible	Infeasible
12	Saturday PM	199.33	0.89	199.33	317
13	Sunday AM	193.67	1.03	193.67	321
14	Sunday PM	175.33	1.14	175.33	318
15	No Restrictions	169.67	5.37	169.67	317

5.0 Conclusion

This research addresses the scheduling of tests for an industrial product that requires in-process and at-completion inspections for market certification. We first introduce a MILP formulation of the problem that determines the optimal schedule of tests that minimizes the make span while ensuring in-process and at-completion inspections are conducted during normal operating hours. This problem can be solved to global optimality in a few seconds using commercial optimization software. Optimizing the schedule for completing the certification testing will assist the company in reducing lead times, accurately estimating delivery dates, and deploying its resources to increase operational efficiency.

Because specialized optimization software can be expensive from both a cost and learning curve perspective, we extend our research and identify a GA-based methodology for solving the problem that produces the same optimal results as the MILP in modest additional run time using Microsoft Excel 2016. We provide results for the MILP and the GA with a series of 15 what-if scenarios demonstrating that the GA is capable of consistently obtaining global optimal solutions in a timely manner. Both of the modeling techniques provide guidance to decision makers concerning the scheduling of product testing at various starting points throughout the work week.

Chapter 5 : Conclusion and Future Research

Conclusion and Future Research

The nature of this research in multi-criteria scheduling related to personnel scheduling creates unique and relevant mathematical optimization models applicable in the business world. We address two applications of employee scheduling using multi-criteria objectives to identify solutions that improve business operations. Additionally, we investigate a scheduling problem from the product testing domain and propose a heuristic solution technique for the problem that is shown to produce very high-quality solutions in short amounts of time.

Chapter 2 addresses a pre-award grant administration workload-to-staff assignment problem. Using a mixed-integer linear programming problem the optimal workload assignment plan is identified, although, requires multiple administrator-to-workload reassignments from the current state to reach optimality. The number of reassignments creates relational cost concern; therefore, we propose a technique to identify the *n* best departmental reassignments from the current state that provides the greatest progress toward the optimal balanced workload solution. The decision maker can make an informed personnel scheduling decision based on the desired reassignment-to-optimality tradeoff.

Chapter 3 addresses a coach-to-program task assignment and tour scheduling problem in a gymnastics facility. We identify a weekly schedule that seeks the most cost-effective set of coach-to-program assignments in a gymnastics facility using an integer linear programming model. The optimal assignment plan differs greatly from the status quo; therefore, we utilize a similar approach from Chapter 2 and use a multiple objective optimization technique to identify the *n* best staff reassignments. This provides the decision maker with a trade-off assessment of the number of program reassignments and the resulting progress achieved toward the optimal staffing cost. The decision maker can make an informed decision about the desired level of disruption (or reassignment) and the resulting impact on staffing cost.

Chapter 4 focuses on product test scheduling in the presence of in-process and at-completion inspection constraints. Such testing arises in the context of the manufacture of products that must perform reliably in extreme environmental conditions. Each product receives a certification at the successful completion of a predetermined series of tests. Operational efficiency is enhanced by determining the optimal order and start times of tests so as to minimize the make span while ensuring that technicians are available when needed to complete in-process and at-completion inspections We first formulate a mixed-integer programming model (MILP) to identify the optimal solution to this problem using IBM ILOG CPLEX Interactive Optimizer 12.7. We also present a genetic algorithm (GA) solution methodology that is implemented and solved in Microsoft Excel. Computational results demonstrate the relative merits of the MILP and GA solution approaches across a number of scenarios. The decision maker can visualize the make span of testing one product at various starting points in the week and make informed product marketing decisions.

The personnel scheduling multiple objective optimization technique to identify the *n* best reassignments from the current state that provides the greatest progress toward the utopian solution can be used in various service domains. Any conflicting objective where the decision maker needs to identify the best number of reassignments (*n*) to choose based on an objective assessment of the relevant trade-offs is applicable.

Exploring the idea of assessing the number of personnel reassignments and analyzing different objective trade-offs while incorporating additional variables opens the research application. For instance, incorporating a measurement for cross-training and analyzing the impact on staffing cost and reassignment while varying the desired level of cross-training would be interesting. Identifying the desired level of cross-training, magnitude of reassignment and operating staffing cost is extremely beneficial to the decision maker.

In relation to the grant application, we can assess how assignments are being made and analyzing where the bottlenecks are in the process. Rather than assigning pre-award grant work based on department we can analyze the impact if grant assignments are based on funding agency (i.e. NSA). We could also assess the bottlenecks and address reassignment at those clustered, overworked points and analyze the impact on the system.

Additionally, investigating different cost or utility functions rather than workload balance or staffing cost allows for further investigation into multi-criteria scheduling. In relation to the gymnastics facility, integrating the idea of penalization (or reward) where task assignments or tour scheduling provided by the employees, manager or customer in relation to the preferred coach would be a relevant application.

Considering the limited research in product test scheduling optimization, it would be valuable to research various products and the testing procedures required. For instance, baby products (i.e. car seats, strollers, baby monitors) must pass a number of tests prior to going to market. Identifying the different processes, for each product may provide an opportunity to not only utilize a version of the genetic algorithm but expand the research into different product testing domains.

Alfares, H. K. (2004). Survey, Categorization, and Comparison of Recent Tour Scheduling Literature. *Annals of Operations Research*, 127(1), 145-175.

Ali, M. F., & Khan, R. Z. (2012). The Study on Load Balancing Strategies in Distributed Computing System. *International Journal of Computer Science and Engineering Survey*, 3(2), 19.

Baker, K. R. (1976). Workforce Allocation in Cyclical Scheduling Problems: A Survey. *Operational Research Quarterly*, 27(1), 155-167.

Barstow, D., Rohde, D., & Saul, S. (2010) Deepwater Horizon's Final Hours, accessed June 20, 2017, [available at http://www.nytimes.com/2010/12/26/us/26spill.html].

Bartels, J. H., & Zimmermann, J. (2009). Scheduling Tests in Automotive R&D Projects. *European Journal of Operational Research*, 193(3), 805-819.

Battaia, O., & Dolgui, A. (2013). A Taxonomy of Line Balancing Problems and Their Solution Approaches. *International Journal of Production Economics*, 142(2), 259-277.

Bechtold, S. E., Brusco, M. J., & Showalter, M. J. (1991). A Comparitive-Evaluation of Labor Tour Scheduling Methods. *Decision Sciences*, 22(4), 683-699.

Becker, C., & Scholl, A. (2006). A Survey on Problems and Methods in Generalized Assembly Line Balancing. *European Journal of Operational Research*, 168(3), 694-715.

Board on Higher Education and Workforce Policy and Global Affairs: Committee on Research Universities (2012). Research Universities and the Future of America: Ten Breakthrough Actions Vital to Our Nation's Prosperity and Security. In N. R. C. o. t. N. Academies (Ed.). Washington D.C.: National Academy of Sciences, 227.

Britt, R. (2012). Universities Report Highest-Ever R&D Spending of $65 Billion in FY 2011. *InfoBrief: National Center for Science and Engineering Statistics*. Arlington, VA: National Science Foundation.

Brucker, P., Qu, R., & Burke, E. (2011). Personnel Scheduling: Models and Complexity. *European Journal of Operational Research*, 210(3), 467-473.

Brusco, M. J., & Johns, T. R. (1998). Staffing a Multiskilled Workforce with Varying Levels of Productivity: An Analysis of Cross-training Policies. *Decision Sciences*, 29(2), 499-515.

Bunte, S., & Kliewer, N. (2009). An Overview on Vehicle Scheduling Models. *Public Transport*, 1(4), 299-317.

Casavant, T. L., & Kuhl, J. G. (1988). A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems. *IEEE Transactions on Software Engineering*, 14(2), 141-154.

Casey Jr., J. J. (2005). Making a Good Thing Even Better. *Research Management Review*, 14(2).

Cattrysse, D. G., & Van Wassenhove, L. N. (1992). A Survey of Algorithms for the Generalized Assignment Problem. *European Journal of Operational Research*, 60(3), 260-272.

Chelst, K., Sidelko, J., Przebienda, A., Lockledge, J., & Mihailidis, D. (2001). Rightsizing and Management of Prototype Vehicle Testing at Ford Motor Company. *Interfaces*, 31(1), 91-107.

Cheng, R., Gen, M., & Tsujimura, Y. (1996). A Tutorial Survey of Job-Shop Scheduling Problems Using Genetic Algorithms-Part I: Representation. *Computers & Industrial Engineering*, 30(4), 983-997.

Cheng, R., Gen, M., & Tsujimura, Y. (1999). A Tutorial Survey of Job-Shop Scheduling Problems Using Genetic Algorithms, Part II: Hybrid Genetic Search Strategies. *Computers & Industrial Engineering*, 36(2), 343-364.

Chou, T. C., & Abraham, J. A. (1982). Load Balancing in Distributed Systems. *IEEE Transactions on Software Engineering*, 8(4), 401-412.

Chun, M. B. J. (2010). Building a Research Administration Infrastructure at the Department Level. *Journal of Research Administration*, 41(3), 77-84.

Dakin, R. J. (1965). A Tree-Search Algorithm for Mixed Integer Programming Problems. *The Computer Journal*, 8(3), 250-255.

Dantzig, G. B. (1954). A Comment on Edie's "Traffic Delays at Toll Booths". *Journal of the Operations Research Society of America*, 2(3), 339-341.

Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science (pre-1986)*, 6(1), 80.

Edie, L. C. (1954). Traffic Delays at Toll Booths. *Operations Research*, 2(2), 107-138.

Evans, G. W. (1984). An Overview of Techniques for Solving Multiobjective Mathematical Programs. *Management Science*, 30(11), 1268-1282.

Fisher, M. L., Jaikumar, R., & Van Wassenhove, L. N. (1986). A Multiplier Adjustment Method for the Generalized Assignment Problem. *Management Science*, 32(9), 1095-1103.

Gamberini, R., Grassi, A., & Rimini, B. (2006). A New Multi-Objective Heuristic Algorithm for Solving the Stochastic Assembly Line Re-Balancing Problem. *International Journal of Production Economics*, 102(2), 226-243.

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The Complexity of Flowshop and Jobshop Scheduling. *Mathematics of Operations Research*, 1(2), 117-129.

Gen, M., Cheng, R., & Lin, L. (2008). Assembly Line Balancing Models. London: Springer London, 477-550.

Ghosh, S., & Gagnon, R. J. (1989). A Comprehensive Literature Review and Analysis of the Design, Balancing and Scheduling of Assembly Systems. *International Journal of Production Research*, 27(4), 637.

Golden, B. L., Raghavan, S., Wasil, E. A., & ebrary, I. (2008). *The Vehicle Routing Problem: Latest Advances and New Challenges*. New York: Springer.

Gomar, J. E., Haas, C. T., & Morton, D. P. (2002). Assignment and Allocation Optimization of Partially Multiskilled Workforce. *Journal of Construction Engineering and Management*, 128(2), 103-109.

Gounaris, A., Yfoulis, C. A., & Paton, N. W. (2012). Efficient Load Balancing in Partitioned Queries Under Random Perturbations. *Association for Computing Machinery: Transactions on Autonomous and Adaptive Systems*, 7(1), 1-27.

Harvey, N. J. A., Ladner, R. E., Lovász, L., & Tamir, T. (2006). Semi-Matchings for Bipartite Graphs and Load Balancing. *Journal of Algorithms*, 59(1), 53-78.

Henderson, W. B., & Berry, W. L. (1977). Determining Optimal Shift Schedules for Telephone Traffic Exchange Operators. *Decision Sciences*, 8(1), 239-255.

Holcomb, D. (2012) 2012 USA Gymnastics Member Club Survey Report, accessed May 7, 2016, [available at https://usagym.org/docs/Education/library/2012_july_16.pdf].

Hwang, C. L., Paidy, S. R., Yoon, K., & Masud, A. S. M. (1980). Mathematical Programming with Multiple Objectives: A Tutorial. *Computers & Operations Research*, 7, 5-31.

Kennedy, C. (2017). Business Owner, Gymnastics. In M. W. Martin (Ed.).

Kim, D. (2013). Task Swapping Networks in Distributed Systems. *International Journal of Computer Mathematics*, 1-23.

Kirby, W. S. (1996). Understanding and Managing Sponsored Research Administration as a System. *Society of Research Administrators Journal*, 27(3/4), 25-35.

Kuhn, H. W. (1955). The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1-2), 83-97.

Kuhn, H. W. (1956). Variants of the Hungarian Method for Assignment Problems. *Naval Research Logistics Quarterly*, 3(4), 253-258.

Land, A. H., & Doig, A. G. (1960). An Automatic Method of Solving Discrete Programming Problems. Evanston: Blackwell Publishing Ltd, 497.

Laporte, G., Mercure, H., & Nobert, Y. (1986). An Exact Algorithm for the Asymmetrical Capacitated Vehicle Routing Problem. *Networks*, 16(1), 33-46.

Lewin, T. (2011) Official Calls for Urgency on College Costs, accessed June, 2016, [available at http://www.nytimes.com/2011/11/30/education/duncan-calls-for-urgency-in-lowering-college-costs.html].

Li, H.-L., Huang, Y.-H., & Fang, S.-C. (2013). A Logarithmic Method for Reducing Binary Variables and Inequality Constraints in Solving Task Assignment Problems. *INFORMS Journal on Computing*, 25(4), 643-653.

Liang, Z., Li, Y., Lim, A., & Guo, S. (2010). Load Balancing in Project Assignment. *Computers and Operations Research*, 37(12), 2248-2256.

Magruder, L. (2013). Director of Pre-Award Administration. In M. W. Martin (Ed.).

Malhotra, M. K., & Ritzman, L. P. (1994). Scheduling Flexibility in the Service Sector: A Postal Case Study. *Production and Operations Management*, 3(2), 100-117.

Marler, R. T., & Arora, J. S. (2004). Survey of Multi-Objective Optimization Methods for Engineering. *Structural and Multidisciplinary Optimization*, 26(6), 369-395.

Martin, A. (2012) Slowly, As Student Debt Rises, Colleges Confront Costs, accessed June 2016, [available at http://www.nytimes.com/2012/05/15/business/colleges-begin-to-confront-higher-costs-and-students-debt.html].

Murty, K. G. (1968). An Algorithm for Ranking All the Assignments in Order of Increasing Cost. *Operations Research*, 16(3), 682-687.

National Center for Science and Engineering Statistics (2015) Higher Education Research and Development Survey (HERD), accessed January 21, 2016, [available at http://ncsesdata.nsf.gov/profiles/site?method=search&n=Virginia].

Neelakantan, P. (2012). Load Balancing in Distributed Systems Using Diffusion Approach. *International Journal of Computer Applications*, 39(4), 1-10.

Oncan, T. (2007). A Survey of the Generalized Assignment Problem and Its Applications. *INFOR*, 45(3), 123-141.

Park, J. S., Lim, B. H., & Lee, Y. (1998). A Lagrangian Dual-Based Branch-and-Bound Algorithm for the Generalized Multi-Assignment Problem. *Management Science*, 44(12-Part-2), S271-S282.

Pedersen, C. R., Relund Nielsen, L., & Andersen, K. A. (2008). An Algorithm for Ranking Assignments Using Reoptimization. *Computers & Operations Research*, 35(11), 3714-3726.

Penmatsa, S., & Chronopoulos, A. T. (2011). Game-Theoretic Static Load Balancing for Distributed Systems. *Journal of Parallel and Distributed Computing*, 71(4), 537-555.

Potts, C. N., & Strusevich, V. A. (2009). Fifty Years of Scheduling: A Survey of Milestones. *The Journal of the Operational Research Society*, 60(S1), 41-68.

Rahimian, E., Akartunali, K., & Levine, J. A Hybrid Integer Programming and Variable Neighbourhood Search Algorithm to Solve Nurse Rostering Problems. *European Journal of Operational Research*, 258(2), 411-423.

Raza, S., Zhu, Y., & Chuah, C.-N. (2011). Graceful Network State Migrations. *IEEE/ACM Transactions on Networking*, 19(4), 1097-1110.

Reich, D., Shi, Y., Epelman, M., Cohn, A., Barnes, E., Arthurs, K., & Klampfl, E. (2016). Scheduling Crash Tests at Ford Motor Company. *Interfaces*, 46(5), 409-423.

Rockwell, S. (2009). The FDP Faculty Burden Survey. *Research Management Review*, 16(2), 29-42.

Ross, G. T., & Soland, R. M. (1975). A Branch and Bound Algorithm for the Generalized Assignment Problem. *Mathematical Programming*, 8(1), 91-103.

Saha, D., Ahmed, A., & Hanumandla, S. (2011). Expectation-Based Efficiency and Quality Improvements in Research Administration: Multi-Institutional Case Studies. *Research Management Review*, 18(2).

Sargent Jr., J. F., Esworthy, R., Gonzalez, H. B., Matthews, C. M., Morgan, D., Moteff, J. D., Schacht, W. H., Smith, P. W., & Upton, H. F. (2012). Federal

Research and Development Funding: FY2012. Congressional Research Service, 45.

Sargent Jr., J. F., Esworthy, R., Matthews, C. M., Morgan, D., Moteff, J. D., Schacht, W. H., Smith, P. W., & Upton, H. F. (2011). Federal Research and Development Funding: FY2011. Congressional Research Service, 49.

Sayın, S., & Karabatı, S. (2007). Assigning Cross-Trained Workers to Departments: A Two-Stage Optimization Model to Maximize Utility and Skill Improvement. *European Journal of Operational Research*, 176(3), 1643-1658.

Sedwick, S. W. (2009). Facilities and Adminstrative Issues: Paying for Adiministration While Under the Cap, Maximizing Recovery, and Communicating Value. *Research Management Review*, 16(2), 22-28.

Shelton, R. N. (2009). Securing the Future for American Research Universities: Keynote Address to the National Council of University Research Administrators Senior Research Leadership Summit. *Research Management Review*. Washington DC, 6-13.

Shi, Y., Reich, D., Epelman, M., Klampfl, E., & Cohn, A. (2017). An Analytical Approach to Prototype Vehicle Test Scheduling. *Omega*, 67, 168-176.

Stanley Jr., S. L., & McCartney, D. A. (2009). Balancing the Burden of Compliance and Faculty Support. *Research Management Review*, 16(2), 14-21.

State Higher Education Executive Officers (2011). State Higher Education Finance FY 2010. Boulder, Colorado, 78.

State Higher Education Executive Officers (2012). State Higher Education Finance FY 2011. Boulder, CO, 79.

State Higher Education Executive Officers (2013). State Higher Education Finance FY 2012. Boulder, Colorado, 81.

State Higher Education Executive Officers (2014). State Higher Education Finance FY 2013. Boulder, Colorado, 81.

Steuer, R. E. (1986). *Mulitple Criteria Optimization: Theory, Computation, and Application*. New York: John Wiley & Sons.

Tessier, B. (2017). Business Owner, Gymnastics. In M. W. Martin (Ed.).

Thompson, G. M. (1990). Shift Scheduling in Services When Employees Have Limited Availability: An L.P. Approach. *Journal of Operations Management*, 9(3), 352-370.

Toth, P., & Vigo, D. (2001). *The Vehicle Routing Problem*. Philadelphia: Society for Industrial and Applied Mathematics.

Toth, P., & Vigo, D. (2014). *Vehicle Routing: Problems, Methods, and Applications*. Philadelphia: Society for Industrial and Applied Mathematics.

Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., & De Boeck, L. (2013). Personnel Scheduling: A Literature Review. *European Journal of Operational Research*, 226(3), 367-385.

Votaw Jr., D. F., & Orden, A. (1951). The Personnel Assignment Problem. *Symposium on Linear Inequalities and Programming, SCOOP 10*. Washington D.C., 155-163.

Yoon, K., & Hwang, C. L. (1995). *Multiple Attribute Decision Making: An Introduction*. Thousand Oaks, [Calif.]: SAGE.

Zhou, S. (1988). A Trace-Driven Simulation Study of Dynamic Load Balancing. *IEEE Transactions on Software Engineering*, 14(9), 1327-1341.

Zülch, G., Rottinger, S., & Vollstedt, T. (2004). A Simulation Approach for Planning and Re-Assigning of Personnel in Manufacturing. *International Journal of Production Economics*, 90(2), 265-277.