Proceedings of The National Conference
On Undergraduate Research (NCUR) 2019
Kennesaw State University
Kennesaw, Georgia
April 11-13, 2019

Airline Crew Scheduling Problem

Matthew McCarver
Department of Mathematics
University of Tennessee at Chattanooga
615 McCallie Avenue
Chattanooga, Tennessee 37403-2598 USA

Abstract

Finding the optimal solution of large scale optimization problems has been problematic for researchers because their computation is often challenging and expensive. A variety of alternative methods are proposed in the literature to make these problems easier to solve, however, the solutions may be infeasible or suboptimal to the original problem. In this paper, we explore the Lagrangian relaxation method and illustrate its use by applying it to the airline crew scheduling problem which is a large scale optimization problem. We examine the feasibility of the solutions obtained by the Lagrangian relaxation method and compare them with the optimal solution. We also propose a heuristic algorithm to generate feasible solutions from infeasible solutions.

1 Introduction and Problem Formulation

Crew scheduling is defined as the problem of assigning a crew to a set of tasks. These problems appear in a number of transportation contexts such as bus and rail transit, truck and rail freight transport, and freight and passenger air transportation. The common aim of all of these is to cover all tasks while minimizing labor costs subject to a wide variety of constraints imposed by safety regulations and labor negotiations. In this study, we use the airline crew assignment problem (ASCP) as the constrained optimization problem to illustrate the Lagrangian relaxation method.

In the ACSP, we are charged with assigning sequences of flight legs (or flight segments) to crews consisting of pilots and flight attendants, stationed in a particular city called a crew base. A flight leg is defined as a flight that takes off in one city, then lands in another. A sequence of flight legs is an ordered arrangement of flight legs. A sequence of flight legs is a feasible sequence of flight legs if it leaves from a crew base and returns to the same base. There is a set of m flight legs, $FL = \{f_1, f_2, \ldots, f_m\}$, and set of n number of flight sequences, $Seq = \{sq_1, sq_2, \ldots, sq_n\}$, where each $sq_j = \{1, 2, \ldots, k_j\}$ is a feasible sequence of flight legs. Let $I = \{1, 2, \ldots, m\}$ and $J = \{1, 2, \ldots, n\}$ be the indices sets for FL and Seq, respectively. The entries $1, 2, \ldots, k_j$ in a set sq_j indicates the order of flights for sequence sq_j for some $j \in J$. Further, each sequence sq_j has an associated cost c_j for some $j \in J$. Let $x \in \mathbb{R}^n$ be the decision variable defined as follows,

$$x_j = \begin{cases} 1 & \text{if the sequence } sq_j \text{ is selected to be scheduled} \\ 0 & \text{otherwise,} \end{cases} \qquad \text{for } j = 1, 2, \ldots, n.$$

The goal of ACSP is to minimize the cost of the k crew assignments that cover all flight legs where k is the number of crew assignments to be chosen. This implies $\sum_{j \in J} x_j = k$. Let the binary coefficient a_{ij} be equal to 1 if a flight leg f_i is in a given sequence sq_j, otherwise a_{ij} is equal to 0. All flight legs in FL must be covered by at least one crew, that is, $\sum_{j \in J} a_{ij} x_j \geq 1$ for all $i \in I$. The mathematical formulation of ACSP is given below.

$$
\begin{aligned}
&\text{Minimize } z = \sum_{j \in J} c_j x_j \\
&\text{Subject to} \\
&\qquad \sum_{j \in J} a_{ij} x_j \geq 1, \qquad i \in I \\
&\qquad \sum_{j \in J} x_j = k \\
&\qquad x_j \in \{0, 1\}, \qquad j \in J.
\end{aligned}
\tag{1}
$$

The model given in (1) represents a 0-1 integer programming problem, and is an extension to classical set covering problem (SCP) with an upper bound constraint $\sum_{j \in J} x_j = k$. The SCP is an NP-hard combinatorial optimization (CO) problem [8] and the upper bound constraint makes it more difficult to solve this problem. This model requires the explicit enumeration of all possible flight sequences. Practical problems often include factors such as labor laws, safety regulations, and personnel availability which make the number of constraints exponential, so finding the optimal solution using the model is difficult or impossible to do.

This issue has been the motivation for a variety of new alternative methods proposed in the literature to obtain a close optimal solution. A well-known technique called Lagrangian relaxation is widely used to find near optimal solutions of many CO problems [5, 6, 4, 12]. In this study we obtain the Lagrangian relaxation of model (1) and propose a heuristic method to find a near optimal solution for ACSP.

In some situations, it is required to include additional conditions and incorporate them into model (1). There are multiple approaches used to solve the crew scheduling problem. Column generation approaches have been successfully implemented to solve large scale scheduling problems by repeatedly generating pairing for a given set partitioning problem [9]. Extensions of this approach have been integrated with heuristic methods such as the heuristic tree search method to solve the crew scheduling problem as a linear programming problem [11]. Meta-heuristic algorithms based on Particle Swarm Optimization that is hybridized with a local search heuristic have been proposed in optimization of crew scheduling [2]. A heuristic based genetic algorithm has been applied to optimizing medium scale airline crew scheduling problems [3], and variable neighborhood search methods have been applied to crew pairing problems [1].

Our work provides two different models associated with ACSP based on Lagrangian relaxation of the ACSP. The Lagrangian relaxation method is applied to linear programming problems where an easy problem to optimize is complicated by certain constraints. In the Lagrangian relaxation method, these complicating constraints are added to the objective function with an assigned Lagrangian multiplier which serves as a penalty weight for violating the constraints. With the relaxation of the complicating constraints, the resulting subproblem is simpler and relatively easier to solve and requires less computational power than the linear programming formulation. This is because the Lagrangian relaxation expands the set of feasible solutions, so the set of feasible solutions to the original linear programming problem are a subset of the set of feasible solutions to the relaxed problem. However, the solutions for the subproblems may not be feasible for the original problem since some constraints are moved to the objective function, allowing them to be violated. We analyze the relationship between the optimal solution of ACSP and the solution provided by the relaxed model. Furthermore we investigate solutions provided by the relaxed model and provide a heuristic algorithm to ensure their feasibility for the ACSP.

Our study is organized as follows. In Section 2, we demonstrate the Lagrangian relaxation model of (1), and discuss the basic properties of Lagrangian relaxation. In Section 3, we illustrate the model by way of example and numerical results. In Section 4 we provide a heuristic algorithm to obtain a feasible and approximately optimal solution to the ACSP. In Section 5 we present our plan for future work based on the model contained herein.

2 Lagrangian Relaxation Method

We explain the Lagrangian relaxation of model (1). The idea of the Lagrangian relaxation is to include complicating constraints as a penalty in the objective function $z = \sum_{j=1}^{n} c_j x_j$. More precisely, we introduce a Lagrangian multiplier vector, $\lambda = (\lambda_1, \ldots, \lambda_m) \in \mathbb{R}_+^m$ for each $\sum_{j \in J} a_{ij} x_j \geq 1$ constraint, and a free Lagrangian multiplier μ for the constraint $\sum_{j \in J} x_j = k$. The Lagrangian multipliers then act as a weight on the constraints. When the constraints and their corresponding Lagrangian multipliers are added to the objective function, if the constraint is violated then the objective function's value increases. With proper selection of the Lagrangian multipliers, the penalties in the objective function deter solutions that are infeasible for the model in (1).

We implement two different Lagrangian relaxation problems, $z_{LR1}(\mathbf{x}, \lambda)$ and $z_{LR2}(\mathbf{x}, \mu)$, of the integer programming problem given in (1). In formulation 1, we penalize the set of constraints $\sum_{j \in J} a_{ij} x_j \geq 1$ and add this set of constraints to the objective function z of the ACSP using Lagrangian multipliers λ_i for $i \in I$. Formulation 1 is expressed as follows:

$$\begin{aligned} &\min z_{\text{LR1}} = \sum_{j \in J} c_j x_j + \sum_{i \in I} \lambda_i \Big(1 - \sum_{j \in J} a_{ij} x_j \Big) \\ &\text{subject to} \\ &\qquad \sum_{j \in J} x_j = k \\ &\qquad x_j \in \{0,1\},\ j \in J \\ &\qquad \lambda_i \geq 0,\ i \in I. \end{aligned} \tag{2}$$

The objective function, $z_{LR1}(\mathbf{x}, \lambda)$, is called the Lagrangian function where

$$z_{LR1}(\mathbf{x}, \lambda, \mu) = \sum_{j \in J} c_j x_j + \sum_{i \in I} \lambda_i \Big(1 - \sum_{j \in J} a_{ij} x_j \Big). \tag{3}$$

In formulation 2, we penalize the equality constraint $\sum_{j \in J} a_{ij} x_j = k$ and add this constraint to the objective function z of the ACSP using the free Lagrangian multiplier μ. Formulation 2 is expressed as follows:

$$\begin{aligned} &\min z_{\text{LR2}} = \sum_{j \in J} c_j x_j + \mu \Big(k - \sum_{j \in J} x_j \Big) \\ &\text{subject to} \\ &\qquad \sum_{j \in J} a_{ij} x_j \geq 1, \qquad i \in I \\ &\qquad x_j \in \{0,1\},\ j \in J \\ &\qquad -\infty \leq \mu \leq \infty. \end{aligned} \tag{4}$$

The objective function, $z_{LR2}(\mathbf{x}, \mu)$, is called the Lagrangian function where

$$z_{LR2}(\mathbf{x}, \lambda, \mu) = \sum_{j \in J} c_j x_j + \mu \Big(k - \sum_{j \in J} x_j \Big). \tag{5}$$

The Lagrangian functions given in (3) and (5) are nonlinear functions. Thus, the models in (2) and (4) are nonlinear optimization problems. But, for a given vector λ or a given μ value, the two models in (2) and (4) can be easily solved compared to model (1).

3 Illustrative Example

In this section we illustrate the model given in (1) and two different Lagrangian relaxation described in Section 2. First, we consider an ACSP given in [7]. Suppose an airline company needs to assign its crews to cover all 11 of its upcoming flights given in Table 1. The airline company has already determined the feasible flight sequences and their associated cost. We will focus on the problem of assigning k number of crews based in San Francisco to flights listed in the first column of Table 1, where each flight requires the same number of crew members. The other 12 columns show the 12 feasible sequences of flights for a crew. (The numbers in each column indicate the order of the flights.) Exactly k number of the sequences need to be chosen (one per crew) in such a way that every flight is covered. (It is permissible to have more than one crew on a flight, where the extra crews would fly as passengers, but union contracts require that the extra crews would still need to be paid for their time as if they were working.) The cost of assigning a crew to a particular sequence of flights (i.e. selecting the flight sequence) is given (in thousands of dollars) in the bottom row of Table 1. The objective is to minimize the total cost of the k crew assignments that cover all the flights.

Table 1: Upcoming flights and feasible flight sequences with their associated costs

	Flights	Sq 1	Sq 2	Sq 3	Sq 4	Sq 5	Sq 6	Sq 7	Sq 8	Sq 9	Sq 10	Sq 11	Sq 12
1	SF to LA	1	0	0	1	0	0	1	0	0	1	0	0
2	SF to Den	0	1	0	0	1	0	0	1	0	0	1	0
3	SF to Sea	0	0	1	0	0	1	0	0	1	0	0	1
4	LA to Chi	0	0	0	2	0	0	2	0	3	2	0	3
5	LA to SF	2	0	0	0	0	3	0	0	0	5	5	0
6	Chi to Den	0	0	0	3	3	0	0	0	4	0	0	0
7	Chi to Sea	0	0	0	0	0	0	3	3	0	3	3	4
8	Den to SF	0	2	0	4	4	0	0	0	5	0	0	0
9	Den to Chi	0	0	0	0	2	0	0	2	0	0	2	0
10	Sea to SF	0	0	2	0	0	0	4	4	0	0	0	5
11	Sea to LA	0	0	0	0	0	2	0	0	2	4	4	2
	cost ($1000's)	2	3	4	6	7	5	7	8	9	9	8	9

The network representation for this problem is given in Figure 1. Each arc denotes a flight and the associated flight number from Table 1.

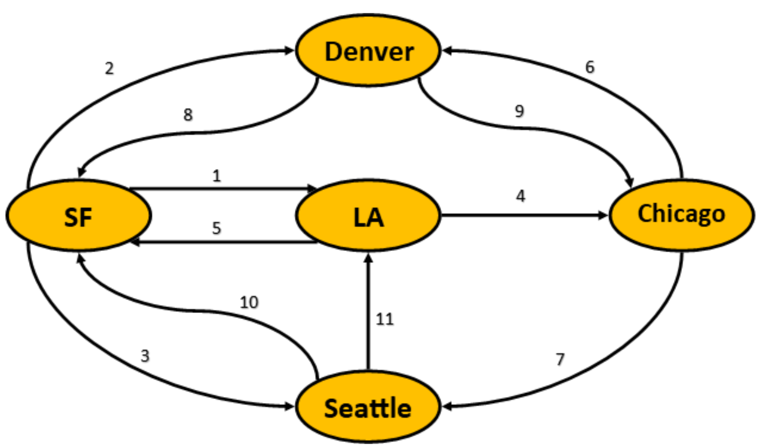

Figure 1: Network representation of the illustrative example

Figure 2 shows the flight sequences 4, 5, 6, and 7 and the corresponding flight legs. For example, flight sequence 4 goes from SF to LA, LA to Chi, Chi to Den, and Den to SF. The ACSP formulation for this problem is

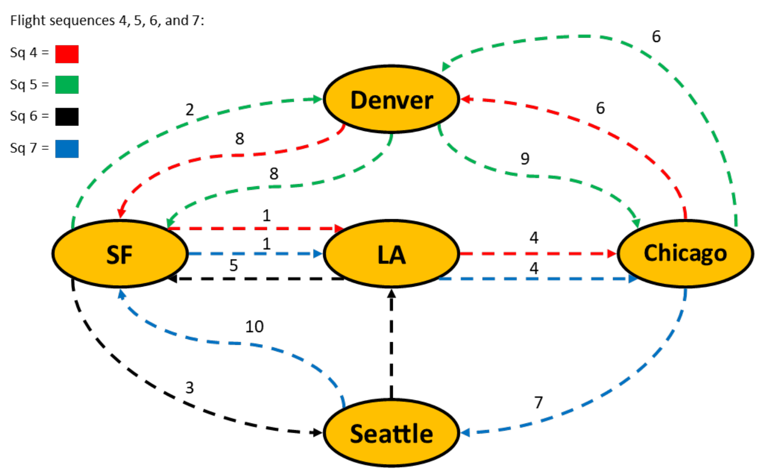

Figure 2: Flight sequences 4, 5, 6, and 7 and the corresponding flight legs

$$\text{minimize } z = 2x_1 + 3x_2 + 4x_3 + 6x_4 + 7x_5 + 5x_6 + 7x_7 + 8x_8 + 9x_9 + 9x_{10} + 8x_{11} + 9x_{12}$$

subject to:

$$\begin{aligned}
x_1 + x_4 + x_7 + x_{10} &\geq 1\ (\lambda_1)\\
x_2 + x_5 + x_8 + x_{11} &\geq 1\ (\lambda_2)\\
x_3 + x_6 + x_9 + x_{12} &\geq 1\ (\lambda_3)\\
x_4 + x_7 + x_9 + x_{10} + x_{12} &\geq 1\ (\lambda_4)\\
x_1 + x_6 + x_{10} + x_{11} &\geq 1\ (\lambda_5)\\
x_4 + x_5 + x_9 &\geq 1\ (\lambda_6)\\
x_7 + x_8 + x_{10} + x_{11} + x_{12} &\geq 1\ (\lambda_7)\\
x_2 + x_4 + x_5 + x_9 &\geq 1\ (\lambda_8)\\
x_5 + x_8 + x_{11} &\geq 1\ (\lambda_9)\\
x_3 + x_7 + x_8 + x_{12} &\geq 1\ (\lambda_{10})\\
x_6 + x_9 + x_9 + x_{10} + x_{11} + x_{12} &\geq 1\ (\lambda_{11})\\
\sum_{j \in J} x_j &= k\ (\mu)\\
x_j \in \{0, 1\},\ \text{for } j \in J
\end{aligned}$$

Since there are 12 flight sequences for this problem, we define 12 binary variables $x = (x_1, x_2, \ldots, x_{12})$. The set of constraints according to model (1) are given below. Let the vector $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{11})$ and μ denote the values of the Lagrangian multipliers that are associated with each constraint.

$$\begin{aligned}
x_1 + x_4 + x_7 + x_{10} &\geq 1\ (\lambda_1)\\
x_2 + x_5 + x_8 + x_{11} &\geq 1\ (\lambda_2)\\
x_3 + x_6 + x_9 + x_{12} &\geq 1\ (\lambda_3)\\
x_4 + x_7 + x_9 + x_{10} + x_{12} &\geq 1\ (\lambda_4)\\
x_1 + x_6 + x_{10} + x_{11} &\geq 1\ (\lambda_5)\\
x_4 + x_5 + x_9 &\geq 1\ (\lambda_6)\\
x_7 + x_8 + x_{10} + x_{11} + x_{12} &\geq 1\ (\lambda_7)\\
x_2 + x_4 + x_5 + x_9 &\geq 1\ (\lambda_8)\\
x_5 + x_8 + x_{11} &\geq 1\ (\lambda_9)\\
x_3 + x_7 + x_8 + x_{12} &\geq 1\ (\lambda_{10})\\
x_6 + x_9 + x_9 + x_{10} + x_{11} + x_{12} &\geq 1\ (\lambda_{11})\\
\sum_{j \in J} x_j &= k\ (\mu)
\end{aligned}$$

According to formulation 1, the corresponding Lagrangian function z_{LR1} is given below:

$$
\begin{aligned}
z_{LR1}(\mathbf{x},\mu) &= 2x_1 + 3x - 2 + 4x_3 + 6x_4 + 7x_5 + 5x_6 + 7x_7 + 8x_8 + 9x_9 + 9x_{10} + 8x_{11} + 9x_{12} \\
&+ \lambda_1\Big(1-(x_1+x_4+x_7+x_{10})\Big) + \lambda_2\Big(1-(x_2+x_5+x_8+x_{11})\Big) + \lambda_3\Big(1-(x_3+x_6+x_9+x_{12})\Big) \\
&+ \lambda_4\Big(1-(x_4+x_7+x_9+x_{10}+x_{12})\Big) + \lambda_5\Big(1-(x_1+x_6+x_{10}+x_{11})\Big) + \lambda_6\Big(1-(x_4+x_5+x_9)\Big) \\
&+ \lambda_7\Big(1-(x_7+x_8+x_{10}+x_{11}+x_{12})\Big) + \lambda_8\Big(1-(x_2+x_4+x_5+x_9)\Big) + \lambda_9\Big(1-(x_5+x_8+x_{11})\Big) \\
&+ \lambda_{10}\Big(1-(x_3+x_7+x_8+x_{12})\Big) + \lambda_{11}\Big(1-(x_6+x_9+x_9+x_{10}+x_{11}+x_{12})\Big).
\end{aligned}
$$

Thus, the formulation 1 of model (1) is given by

$$
\begin{aligned}
&\text{Minimize } z_{LR1} \\
&\text{Subject to} \\
&\qquad \sum_{j\in J} x_j = k \\
&\qquad x_j \in \{0,1\}, \qquad j \in J.
\end{aligned} \tag{6}
$$

According to formulation 2, the corresponding Lagrangian function z_{LR2} is given below:

$$
z_{LR1}(\mathbf{x},\lambda,\mu) = 2x_1 + 3x_2 + 4x_3 + 6x_4 + 7x_5 + 5x_6 + 7x_7 + 8x_8 + 9x_9 + 9x_{10} + 8x_{11} + 9x_{12} + \mu\Big(k - \sum_{j\in J} x_j\Big).
$$

Thus, the formulation 2 of model (1) is given by

$$
\begin{aligned}
&\text{Minimize } z_{LR2} \\
&\text{Subject to} \\
&\qquad \sum_{j\in J} a_{ij}x_j \geq 1 \\
&\qquad x_j \in \{0,1\}, \qquad j \in J.
\end{aligned} \tag{7}
$$

4 Numerical Work

This section summarizes the numerical experiments using several test cases. The model given in (1) and the Lagrangian relaxation formulations 1 and 2 were implemented in MATLAB, version R2019a [10]. Since the functions z_{LR1} and z_{LR2} are nonlinear functions, we solved the resulting formulations using the fmincon routine of MATLAB. This routine finds a constrained minimum of a several variables function starting at an initial estimate using a sequential quadratic programming (SQP) method. In order to compare the solutions provided by model (1) with relaxed solutions provided by Lagrangian formulations, the 'intlinprog' solver in MATLAB was used to solve the model (1). The comparison of model (1) and the two Lagrangian formulations was conducted using different k values from the test problem given in Section 3.

Table 2 and 3 provide the result of this analysis. Let x^* be the optimal solution of model (1). Let $\bar{x}$ and $\hat{x}$ be the solutions provided by fmincon routine for formulations 1 and 2, respectively. The values of the objective functions z evaluated at $\bar{x}$ and $\hat{x}$ are shown in last two columns of Table 2.

We use Table 3 to discuss the violation of constraints under each formulation with respect to the relaxed solutions $\bar{x}$ and $\hat{x}$, respectively. With formulation 1, we allow violation of coverage constraints $\sum_{j\in J} a_{ij}x_j \geq 1$ for $i \in I$ and Block 1 of Table 3 shows outcome of this analysis. For example, first row of Table 3 shows the coverage of each flight leg (denoted by FL1, FL2,. . . , FL11) leg when $k = 3$ with respect to formulation 1. We observe that the solution $\bar{x}$ provides at least one coverage for all flight legs except for FL6, FL8, and FL10. Thus we are missing 27.27% of flight legs to be covered with this solution. Similarly with $k = 4$ and $k = 5$, we observe that 27.27% and 18.18% coverages are missing with the relaxed solution $\bar{x}$, respectively. Although this solution doesn't cover all the flight legs, it satisfies the required number of crew-assignments.

With formulation 2, we allow violation of coverage constraints $\sum_{j\in J} x_j = k$ and Block 2 of Table 3 shows outcome of this analysis. For example, third to the last row of Table 3 shows the coverage of each flight leg is satisfied. But we observe that the solution $\hat{x}$ does not satisfy the constraint $\sum_{j\in J} x_j = k$ for $k = 4$ and $k = 5$. This solution provides the coverage for each flight leg, but we are missing 1 and 2 crews with the relaxed solution $\hat{x}$ when $k = 4$ and $k = 5$, respectively.

Table 2: Optimal solution of model in (1) and solutions provided by Formulations 1 and 2

k	Optimal solution x^*												Formulation 1: Relaxed solution $\bar{x}$												$z(x^*)$	$z(\bar{x})$
3	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	1	0	18	22
4	1	0	1	1	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	20	24
5	1	1	1	1	0	0	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	1	0	0	23	27
k	Optimal solution x^*												Formulation 2: Relaxed solution $\hat{x}$												$z(x^*)$	$z(\hat{x})$
3	1	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	1	18	18
4	1	0	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	20	18
5	1	1	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	23	18

Table 3: Infeasiblity level of solutions provided by Formulations 1 and 2

Block 1 for Formulation 1												
	Formulation 1: Coverage Relaxed solution $\bar{x}$											In-feasibility Level
k	FL1	FL2	FL3	FL4	FL5	FL6	FL7	FL8	FL9	FL10	FL11	
3	1	1	1	1	3	0	2	0	1	0	3	27.27%
4	4	0	0	3	2	1	2	1	0	1	1	27.27%
5	3	2	0	2	2	2	1	3	1	0	1	18.18 %
Block 2 for Formulation 2												
	Formulation 2: Coverage Relaxed solution $\hat{x}$											In-feasibility Level
k	FL1	FL2	FL3	FL4	FL5	FL6	FL7	FL8	FL9	FL10	FL11	
3	1	1	1	1	1	1	1	1	1	1	1	0
4	1	1	1	1	1	1	1	1	1	1	1	one crew is missing
5	1	1	1	1	1	1	1	1	1	1	1	two crews are missing

Figure 3 shows the comparison of the objective function z of model (1) with respect to the solutions $x^*, \bar{x}$ and $\hat{x}$ for $k = 2, 3, 4$, respectively. We observe that $z(\hat{x}) \le z(x^*) \le z(\bar{x})$. Further, we observe that $z(x^*) = z(\hat{x})$ for $k = 3$. Even though $z(\hat{x}) \le z(x^*)$ for $k = 4$ and $k = 5$, $\hat{x}$ is not a feasible solution for model (1).

5 Heuristic Algorithm

We make the solution $\hat{x}$ of formulation 2 return to a feasible solution for model (1). First, we obtain the formulation 2 of the ACSP and obtain the solution $\hat{x}$ of formulation 2. Then we identify the variables that are not selected for the solution $\hat{x}$. Let the set US denote the unselected variables. We define a preferred variable from the set US as below:

Definition 1: A variable x_j in US is preferred over the other variables in US if its cost (that is c_j) is less than or equal to all other costs associated with the variables in US.

We identify a preferred variable from US and add it to the solution $\hat{x}$. Then we remove that variable from US. If the new solution is a feasible solution for model (1) we stop the procedure. Otherwise, we continue the procedure until we obtain a feasible solution for model (1).

We explain this procedure using the example from the previous section and the results are given in Table 4. When $k = 4$ and $k = 5$, the set $US = \{1, 2, 5, 6, 7, 8, 9, 10, 12\}$. We need to add one variable to get a feasible solution when $k = 4$ since we are missing one crew. Thus we add the variable x_1 since it is the preferred variable according to Definition 1. The cost of x_1 is 2. The new solution (so called the heuristic solution) is denoted by $\hat{\hat{x}}$, and $z(\hat{\hat{x}}) = z(\hat{x}) + c_2 = 18 + 2 = 20$. Similarly, we need to add two variables to get a feasible solution when $k = 5$ since we are

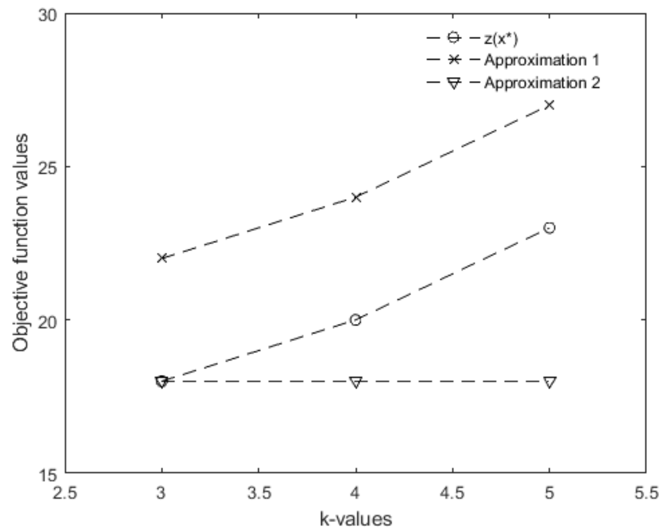

Figure 3: Comparison of objective function values at solutions x^*, $\bar{x}$ and $\hat{x}$

missing two crews. Thus we add the variable x_1 first since it is the preferred variable according to Definition 1 and the variable x_2 second since it is the preferred variable according to Definition 1. The costs of x_1 and x_3 are 2 and 3, respectively. Thus $z(\hat{\hat{x}}) = z(\hat{x}) + c_2 + c_3 = 18 + 2 + 3 = 23$. We observe that this heuristic algorithm provides the objective values same as the optimal solution. This may not be the case in general since the heuristic algorithm is used to return a feasible solution for model (1) that is approximately equal to the optimal solution of model (1). For larger linear programming problems, the heuristic algorithm may return feasible, but suboptimal, solutions for model (1).

Table 4: Optimal objective values and objective values evaluated at the heuristic solution $\hat{\hat{x}}$

k	Formulation 2: Relaxed solution $\hat{x}$												$z(x^*)$	$z(\hat{x})$	$z(\hat{\hat{x}})$
cost	2	3	4	6	7	5	7	8	9	9	8	9			
3	1	0	0	0	1	0	0	0	0	0	0	1	18	18	
4	0	0	1	1	0	0	0	0	0	0	1	0	20	18	20
5	0	0	1	1	0	0	0	0	0	0	1	0	23	18	23

6 Discussion and Future Work

In this paper, we explored two formulations of the ACSP as Lagrangian relaxation problems. We proposed a heuristic algorithm to tackle the problem of infeasible solutions being generated by the Lagrangian relaxation of the ACSP by transforming infeasible solutions provided by the second Lagrangian relaxation formulation into feasible solutions for the ACSP. Computational results based on a small data set show that our heuristic algorithm works well and provides the optimal solution for the ACSP.

Our future study involves searching for a real data set to apply the Lagrangian relaxation and our heuristic algorithm to and test the optimality of our methods. Furthermore, we will investigate advanced searching methods to improve our current heuristic algorithm.

References

[1] Alba Agustin, Angel Juan, and Eduardo G Pardo. A variable neighborhood search approach for the crew pairing problem. *Electronic Notes in Discrete Mathematics*, 58:87–94, 2017.

[2] Ali Azadeh, M Hosseinabadi Farahani, H Eivazy, S Nazari-Shirkouhi, and G Asadipour. A hybrid meta-heuristic algorithm for optimization of crew scheduling. *Applied Soft Computing*, 13(1):158–164, 2013.

[3] Nihan Çetin Demirel and Muhammet Deveci. Novel search space updating heuristics-based genetic algorithm for optimizing medium-scale airline crew pairing problems. *International Journal of Computational Intelligence Systems*, 10(1):1082–1101, 2017.

[4] Matteo Fischetti and Paolo Toth. An additive bounding procedure for combinatorial optimization problems. *Operations Research*, 37(2):319–328, 1989.

[5] Marshall L Fisher. The lagrangian relaxation method for solving integer programming problems. *Management science*, 27(1):1–18, 1981.

[6] Marshall L Fisher. The lagrangian relaxation method for solving integer programming problems. *Management science*, 50(12_supplement):1861–1871, 2004.

[7] Frederick S Hillier. *Introduction to Operations Research: Frederick S. Hillier and Gerald J. Lieberman*. Holden-Day, 1967.

[8] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[9] Diego Klabjan, Ellis L Johnson, George L Nemhauser, Eric Gelman, and Srini Ramaswamy. Solving large airline crew scheduling problems: Random pairing generation and strong branching. *Computational Optimization and Applications*, 20(1):73–91, 2001.

[10] The Mathworks, Inc., Natick, Massachusetts. *MATLAB version (R2019a)*, 2019.

[11] Meinolf Sellmann, Kyriakos Zervoudakis, Panagiotis Stamatopoulos, and Torsten Fahle. Crew assignment via constraint programming: integrating column generation and heuristic tree search. *Annals of Operations Research*, 115(1-4):207–225, 2002.

[12] Yuanhui Zhang, Peter B Luh, Kiyoshi Yoneda, Toshiyuki Kano, and Yuji Kyoya. Mixed-model assembly line scheduling using the lagrangian relaxation technique. *Iie Transactions*, 32(2):125–134, 2000.