Computers & Industrial Engineering 99 (2016) 106–111

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Prediction-based relaxation solution approach for the fixed charge network flow problem

Weili Zhang, Charles D. Nicholson *

School of Industrial and Systems Engineering, University of Oklahoma, USA

ARTICLE INFO

Article history:
Received 1 February 2016
Received in revised form 23 May 2016
Accepted 14 July 2016
Available online 15 July 2016

Keywords:
Network optimization
Fixed charge network flow
Heuristics

ABSTRACT

A new heuristic procedure for the fixed charge network flow problem is proposed. The new method leverages a probabilistic model to create an informed reformulation and relaxation of the FCNF problem. The technique relies on probability estimates that an edge in a graph should be included in an optimal flow solution. These probability estimates, derived from a statistical learning technique, are used to reformulate the problem as a linear program which can be solved efficiently. This method can be used as an independent heuristic for the fixed charge network flow problem or as a primal heuristic. In rigorous testing, the solution quality of the new technique is evaluated and compared to results obtained from a commercial solver software. Testing demonstrates that the novel prediction-based relaxation outperforms linear programming relaxation in solution quality and that as a primal heuristic the method significantly improves the solutions found for large problem instances within a given time limit.

1. Introduction

The fixed charge network flow problem (FCNF) can be described on a network $G = (N, A)$, where N and A are the sets of nodes and edges, respectively. Each node $i \in N$ has a supply/demand commodity requirement R_i ($R_i > 0$ if node i is a supply node; $R_i < 0$ if node i is a demand node; otherwise, $R_i = 0$). Each edge $(i,j) \in A$ has costs associated with commodity flow. Let c_{ij} and f_{ij} denote the variable and fixed costs, respectively. An artificial capacity value, M_{ij}, can be used in the problem formulation to ensure that the fixed cost f_{ij} is incurred whenever there is a positive flow on $(i,j) \in A$. There are two types of variables in the FCNF, edge (i,j) flow and usage, denoted as x_{ij} and y_{ij}, respectively. The latter is a binary variable representing the decision to use $(i,j) \in A$ for routing commodities and incurs the fixed cost f_{ij}. The formulation is provided in Eqs. (1)–(4).

$$\min \sum_{(i,j)\in A} (c_{ij}x_{ij} + f_{ij}y_{ij}) \quad (1)$$

$$\text{s.t.} \sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = R_i \quad \forall i \in N \quad (2)$$

$$0 \leqslant x_{ij} \leqslant M_{ij}y_{ij} \qquad \forall (i,j) \in A \quad (3)$$

$$y_{ij} \in \{0,1 \qquad \forall (i,j) \in A \quad (4)$$

The objective function in (1) is the sum of variable and fixed costs incurred for the solution. Constraint (2) ensures that the inflow and outflow satisfy the supply or demand requirements at node $i \in N$. Constraint (3) creates a logical relationship between x_{ij} and y_{ij}. Constraint (4) defines y_{ij} as binary, which makes the problem a mixed binary programming problem.

Many practical problems can be modeled as a FCNF problem or variation thereof, such as transportation problems (El-Sherbiny & Alhamali, 2013), the lot sizing problem (Steinberg & Napier, 1980), the facility location problem (Melo, Nickel, & Saldanha-da Gama, 2009; Nozick, 2001), network design (Costa, 2005; Ghamlouche, Crainic, & Gendreau, 2003; Lederer & Nambimadom, 1998) and others (Armacost, Barnhart, & Ware, 2002; Jarvis, Rardin, Unger, Moore, & Schimpeler, 1978). The FCNF problem is an NP-hard problem and over the decades, a significant quantity of research has been directed towards providing solution approaches to the FCNF. Many techniques utilize variations of the branch-and-bound (B&B) algorithm to search for the exact solution of the FCNF (Barr, Glover, & Klingman, 1981; Cabot & Erenguc, 1984; Driebeek, 1966; Hewitt, Nemhauser, & Savelsbergh, 2010; Kennington & Unger, 1976; Ortega & Wolsey, 2003; Palekar, Karwan, & Zionts, 1990). The B&B algorithm may be inefficient due to the lack of the tight bounds during the linear programming (LP) relaxation.

State-of-the-art MIP solvers combine a variety of cutting plane strategies, heuristic techniques with B&B to search for the exact optimal solution. Modern MIP solvers use preprocessing methods to reduce the search space and significantly accelerate the solving

* Corresponding author at: School of Industrial and Systems Engineering, University of Oklahoma, 202 W. Boyd St., Room 124, Norman, OK, USA.
E-mail address: cnicholson@ou.edu (C.D. Nicholson).

http://dx.doi.org/10.1016/j.cie.2016.07.014
0360-8352/

processes (Bixby, Fenelon, Gu, Rothberg, & Wunderling, 2000). More details of the preprocessing techniques can be found in Nemhauser and Wolsey (1988), Wolsey (1998), Fügenschuh and Martin (2005), Mahajan (2010). Additionally, due to the impractical computational effort to obtain exact solutions for large instances, heuristic approaches to obtain near-optimal solutions have generated considerable research interest (Adlakha & Kowalski, 2010; Antony Arokia Durai Raj & Rajendran, 2012; Balinski, 1961; Kim & Pardalos, 1999; Molla-Alizadeh-Zavardehi, Hajiaghaei-Keshteli, & Tavakkoli-Moghaddam, 2011; Monteiro, Fontes, & Fontes, 2011; Sun, Aronson, McKeown, & Drinka, 1998).

In this study, a novel *prediction-based relaxation* (PBR) method is proposed. PBR may be used as an FCNF heuristic solution technique or as a primal heuristic to potentially improve on exact search solution quality and time. The new method leverages a probabilistic model to create an informed reformulation and relaxation of the FCNF problem. For the FCNF problem, a binary probabilistic classification model is required. Approaches to develop such models are common in the field of statistical learning and include logistic regression, linear discriminant analysis, partial least squares discriminant analysis, and random forests among others. The present work demonstrates the utility of PBR by implementing one such model, a logistic regression developed by Nicholson and Zhang (2016).

The primary contribution of this work is to introduce and evaluate a new paradigm for approaching the classical FCNF problem. In particular, elements from statistical learning are combined with more traditional solution techniques from Operations Research. Section 2 provides a brief review of the predictive model from Nicholson and Zhang (2016). Subsequently, the necessary mathematical transformations to leverage the predictive model results are described and the formal PBR mathematical model is presented. Section 3 describes the experiments to evaluate PBR as an independent heuristic and explores the strategy to combine PBR and the B&B algorithm as an exact solution approach. The work is concluded in Section 4.

2. Prediction-based relaxation

2.1. Motivation

The prediction-based relaxation approach is dependent on an underlying statistical model to produce probabilities for positive edge flow in an FCNF instance. A variety of predictive modeling methods to produce such probabilities are possible. One such possibility is the statistical learning model developed in Nicholson and Zhang (2016). Their model is based on the 28 variables described in Table 1. The authors classify these variables as being associated with four types of network characteristics: overall network level (e.g., total number of nodes), edge specific attributes (e.g., variable cost), linear relaxation based variables (e.g., edge flow in an relaxed problem), and lastly, variables related to the nodes incident to an edge (e.g., node degree). While many network and network component features are possible, their guiding principle was to find predictive variables that are relatively straightforward and easy to compute since their goal was to introduce and explore a new paradigm for understanding and exploiting FCNF problems. Through the use of initial trial and error tests and secondly, the use of Akaike information criterion informed stepwise regression, this small subset was selected. These features are used as predictors in a logistic regression model to predict components of the FCNF optimal solution. The associated regression coefficients are provided in Table 1. In particular, the model produces a likelihood of edge usage with a logistic regression model. The response variable is a binary variable, indicating whether or not the edge has positive flow in the optimal solution. Let Y_{ij} denote the response variable,

$$Y_{ij} = \begin{cases} 1, & \text{edge } (i,j) \text{ is used in the FCNF optimal solution} \\ 0, & \text{otherwise.} \end{cases}$$

Table 1
Logistic regression model from Nicholson and Zhang (2016).

Predictor description	Predictor notation	Regression coefficient
	(Intercept)	8.32
Number of nodes	n	−0.048
Number of edges	m	−0.0055
Variable cost on (i,j)	c_{ij}	−0.0879
Fixed cost on (i,j)	f_{ij}	−0.0002
Fixed to variable cost ratio	γ_{ij}	<0.0001
Tail node type: transhipment	$t_i = 0$	−0.543
Tail node type: supply	$t_i = 1$	2.17
Head node type: transhipment	$t_j = 0$	−3.01
Head node type: supply	$t_j = 1$	−2.38
Outdegree of tail node	$\bar{d}_{i*}$	−2.59
Indegree of tail node	$\bar{d}_{*i}$	−2.69
Outdegree of head node	$\bar{d}_{j*}$	−2.3
LP relaxation usage	$\bar{l}^B_{ij}$	1.4
LP relaxation flow	$\bar{l}_{ij}$	5.76
Tail node requirements	$\bar{r}_i$	0.967
Tail adjacent in-supply	$\bar{r}^S_{i*}$	0.765
Tail adjacent in-demand	$\bar{r}^D_{i*}$	0.878
Tail adjacent out-supply	$\bar{r}^S_{*i}$	−0.215
Tail adjacent out-demand	$\bar{r}^D_{*i}$	0.464
Head adjacent in-supply	$\bar{r}^S_{j*}$	−0.797
Head adjacent in-demand	$\bar{r}^D_{j*}$	−0.917
Head adjacent out-supply	$\bar{r}^S_{*j}$	−0.122
Demand head nodes adjacent to tail	$\bar{d}^D_{i*}$	−1.3
Supply tail nodes adjacent to tail	$\bar{d}^S_{*i}$	1.84
Demand tail nodes adjacent to tail	$\bar{d}^D_{*i}$	4.57
Demand head nodes adjacent to head	$\bar{d}^D_{j*}$	−1.58
Supply tail nodes adjacent to head	$\bar{d}^S_{*j}$	3.19
Demand tail nodes adjacent to head	$\bar{d}^D_{*j}$	1.26

Logistic regression is supervised learning technique to develop a probabilistic classifier. The resulting model, based on the input data, assigns a probability that the response variable attains a given value. PBR requires a probabilistic classifier, albeit it is in no way limited to logistic regression derived models. Ideally, a probabilistic model for PBR is both highly accurate and efficient to implement. The logistic regression model conceived and studied in Nicholson and Zhang (2016) meets both of these criteria. Ideally, if edges used in an optimal FCNF solution could be predicted perfectly, then the B&B search would be altogether eliminated. Perfect predictions however are not a reasonable aspiration in general. If, however, a predictive model were highly accurate, then at least the search space could potentially be reduced or the search otherwise informed. The goal of PBR is to exploit such probability models.

2.2. Mathematical model

In this work, the probability from the final logistic regression model is used to develop a novel heuristic for the FCNF. Let $p_{ij} \approx P(Y_{ij} = 1)$ denote the probability estimate that edge $(i,j) \in A$ is used in the FCNF problem. The value for p_{ij} is derived from any appropriate predictive model. Let $c'_{ij} = -\ln p_{ij}$ where $0 < p_{ij} \leqslant 1\ \forall (i,j) \in A$. The PBR cost function is denoted by z'_{PBR} and defined as

$$z'_{\text{PBR}} = \sum_{(i,j)\in A} c'_{ij} x_{ij} = \sum_{(i,j)\in A} -x_{ij} \ln p_{ij} = -\ln \prod_{(i,j)\in A} p_{ij}^{x_{ij}}$$

The PBR problem formulation is: $\min z'_{PBR}$ subject to (2)–(4). PBR is linear and can be solved efficiently. Intuitively, in a simple feasible instance with single supply node s, single demand node t, and total network supply equal to 1, the PBR solution will be a *most probable feasible path* from s to t. That is, if P_{st} denotes a most probable feasible path from s to t, then

$$z'_{PBR} = -\ln \prod_{(i,j)\in P_{st}} p_{ij}$$

is minimal and $x_{ij} = 1 \iff x_{ij} \in P_{st}$. In general, for feasible FCNF problems with $\mathbf{x} \geqslant 0$, the solution to PBR identifies a set of edges which form feasible likely paths from possibly many supply nodes to many demand nodes.

Let x_{ij}^{PBR} denote the value of optimal flows on edge $(i,j) \in A$ in the PBR problem. Let y_{ij}^{PBR} be a binary variable corresponding to edge usage in the PBR problem,

$$y_{ij}^{PBR} = \begin{cases} 1, & x_{ij}^{PBR} > 0 \\ 0, & \text{otherwise} \end{cases} \quad \forall (i,j) \in A.$$

The optimal solution of PBR provides a feasible solution to the original FCNF problem since $\mathbf{x}^{PBR}$ and $\mathbf{y}^{PBR}$ satisfy constraints (2)–(4). Let z_{PBR} denote the objective value from (1) computed from the PBR derived solution,

$$z_{PBR} = \sum_{(i,j)\in A} \left(c_{ij} x_{ij}^{PBR} + f_{ij} y_{ij}^{PBR} \right).$$

3. Computational results

3.1. Experimental design

In order to evaluate the solution quality and efficiency of PBR as a heuristic approach to the FCNF, 900 problem instances are generated which encompass a variety of network sizes and densities. The tests are grouped into three difficulty levels based on problem size, namely the number of nodes: 300 small, "easy" problems (with U(10, 300) nodes), 300 medium-sized and "medium" difficulty problems (with U(350, 650) nodes), and 300 larger, "hard" problems (with U(700, 1000) nodes) are created. The number of edges, m, is also generated in a stochastic manner. In particular, a value for $\frac{m}{2}$ is uniformly randomly chosen such that $n - 1 \leqslant \frac{m}{2} \leqslant \frac{n(n-1)}{2}$. The first $n - 1$ edges are added to the graph to ensure connectivity as follows. Assuming the nodes in the graph are identified with a unique index, $i = 1, 2, \ldots, n$, then for every node $i > 1$, an edge (i, k) is added to the instance where $k < i$ is a randomly selected node index. The remaining $\frac{m}{2} - n + 1$ quantity of edges is added to the graph randomly. Each of the $\frac{m}{2}$ undirected edges is replaced by two directed edges.

The percentage of supply, demand and transshipment nodes are randomly chosen with respectively approximate probabilities, 0.2, 0.2, and 0.6. The probabilities are approximate because adjustments are necessary to ensure the instance is feasible. The total number of supplies is randomly assigned on U(1000, 2000). Let α_s and α_d denote the percentage of supply and demand nodes, respectively. All instances have relatively high fixed to variable cost ratios denoted by γ. The variable and fixed costs are randomly assigned on U(0, 10) and U(20000, 60000), respectively.

Table 2 reports statistics (minimum, first quartile, median, mean, third quartile, and maximum) for the number of nodes, edges, network density ρ, average supply per node $\bar{S}$, fixed to variable cost ratio, and percentages of supply and demand nodes in the network instances. The smallest instance has 10 nodes and 20 directed edges, whereas the largest problem contains 1000 nodes and 82,010 edges.

Table 2
Test set instance characteristics.

Parameters	Min	Q1	Median	Mean	Q3	Max
n	10	250	400	446.7	700	1000
m	20	9497	26270	29520	46480	82010
ρ	0.03	0.24	0.58	0.55	0.89	1.00
$\bar{S}$	333	858.2	947	968.4	1039	1971
γ	7110	7986	8000	8008	8010	9809
α_s	0.100	0.300	0.318	0.317	0.333	0.500
α_d	0.080	0.189	0.200	0.203	0.216	0.400

The empirical analysis goal is threefold: (i) compare PBR solution quality and time with LP relaxation, (ii) evaluate PBR as an initial solution approach against a state-of-the-art commercial software, and (iii) evaluate a possible hybrid strategy whereby PBR is used to inject solutions into an exact search. The first of the two analysis goals are realized in Section 3.2. The hybrid strategy is detailed in Section 3.3.

Gurobi 6.0 software is used as the optimization software. By default, Gurobi uses 14 different MIP heuristics, 16 cutting plane strategies, and several presolve techniques (Optimization, 2012). The best objective value found using Gurobi 6.0 is denoted as z_{GRB}. The objective value of the FCNF found using LP relaxation is denoted by z_{LP}. The experiments are performed on a Windows 7 64 bit machine with Intel Xeon CPU E5-1620 and 8 GB RAM. The time limit for all three techniques is 60 s. If the default Gurobi software cannot obtain a feasible solution within 60 s, the problems instances are excluded from the detailed analysis in Section 3.2 since no comparisons are possible. From the initial 900 generated test instances, 274 are excluded based on this principle. Therefore, the statistical analysis is performed on the remaining 626 cases: 224 easy, 236 medium, and 166 hard problems.

3.2. Results analysis: initial solutions

Let z_{gap}^X denote the percentage gap between z_{PBR} and z_X,

$$z_{gap}^X = \frac{z_{PBR} - z_X}{|z_X|} \times 100\%, \qquad X \in \{LP, GRB\}. \tag{5}$$

Performance regarding solution quality is measured with respect to z_{gap}^X. If PBR improves on the LP (or GRB) solution, the value of z_{gap}^{LP} (or z_{gap}^{GRB}) is negative. The mean gap percentages and the associated p-values from a paired t-test are listed in Table 3. The distribution of observed gap values with the LP solution and the commercial solver solution by problem difficulty are depicted in Figs. 1 and 2. The PBR solution is consistently better than the LP relaxation solution across all three difficulty levels; overall PBR finds solutions that are 32% better than the LP solution. As the problem difficulty increases, the performance gap increases, obtaining a 36% average improvement for the hardest problems. PBR also improves on the solution quality when compared to the commercial solver during the first 60 s of computation time. The overall improvement is a statistically significant 6% improvement. Again, the performance benefits from PBR improve as the problem difficulty increases – the

Table 3
Mean objective gap percentage.

	Levels			
	Easy	Medium	Hard	Overall
z_{gap}^{LP}	−25.23% (<0.001)	−35.57% (<0.001)	−36.18% (<0.001)	−32.03% (<0.001)
z_{gap}^{GRB}	14.29% (<0.001)	−1.25% (0.004)	−29.84% (<0.001)	−6.16% (<0.001)

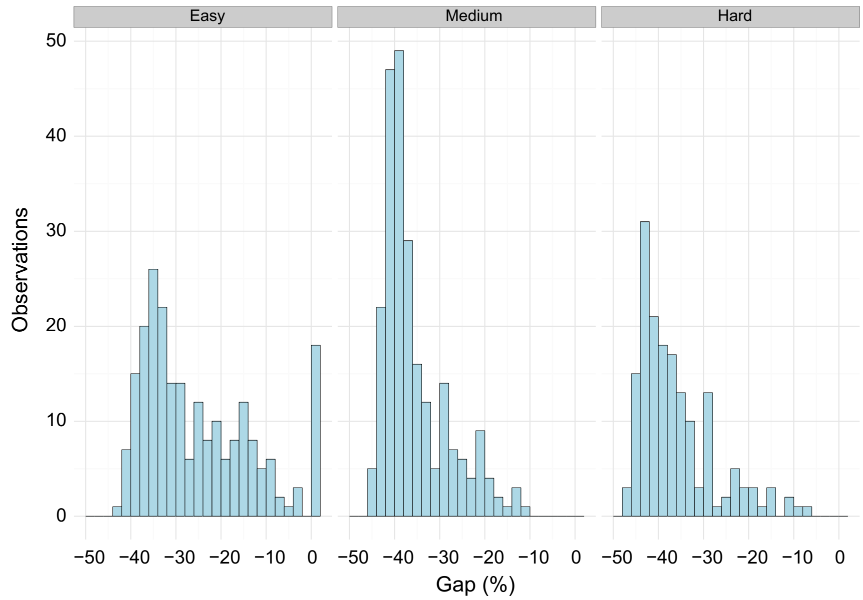

Fig. 1. Distribution of gap between z_{PBR} and z_{LP} by level.

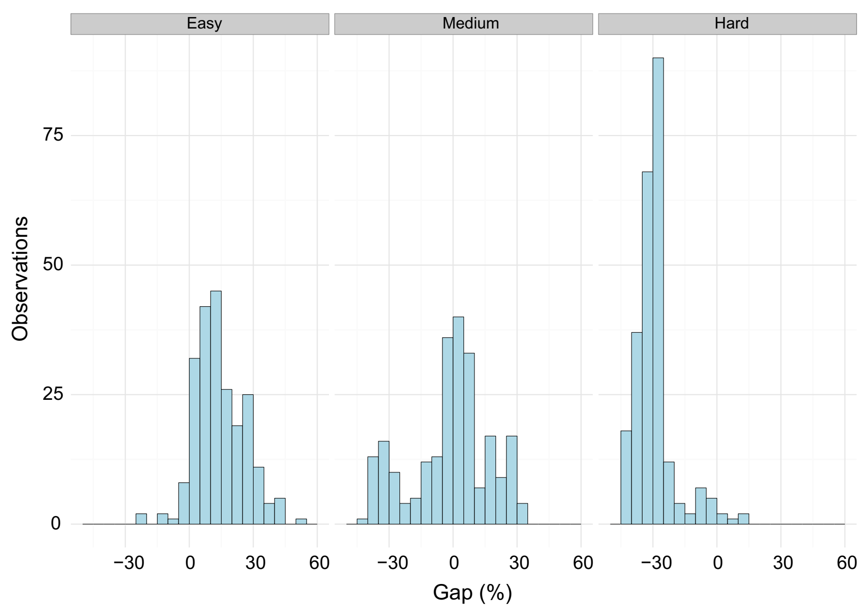

Fig. 2. Distribution of gap between z_{PBR} and z_{GRB} by level.

easiest problems do not benefit from PBR, while the hardest problems are improved by nearly 30%. Among the hard FCNF problems, PBR outperforms the commercial solver in 98% of the instances regarding solution quality and has an average computation time of only 0.33 seconds.

The PBR solution time (computation of features, calculation of probabilities, problem revision and solution, and computation of objective value) is primarily dominated by the LP relaxation solution time necessary for constructing two predictors in the logistic regression model. The LP relaxation solution time in seconds is plotted against the number of edges in the network in Fig. 3. The figure also depicts the *incremental* PBR computation time. For relatively smaller problems (e.g., less than 5000 edges), both the LP solution time and the incremental PBR time are small. As the problem size increase however, the LP solution time becomes a significant factor in computation, whereas the incremental PBR computation time appears to be constant with respect to the network size.

Given the solution quality of PBR and the relatively small incremental computational time, PBR is a candidate technique to improve practical solution efficiency of FCNF problems. Section 3.3 provides a description, implementation, and test for using PBR as a primal heuristic.

3.3. Results analysis: PBR hybrid approach

Primal heuristics are applied at the root node or early in the B&B process in order to find good integer feasible solutions. Examples include the rounding method (Goemans & Williamson, 1995), relaxation enforced neighborhood search (Berthold, 2007), and feasibility pump (Achterberg & Berthold, 2007; Bertacco, Fischetti, &

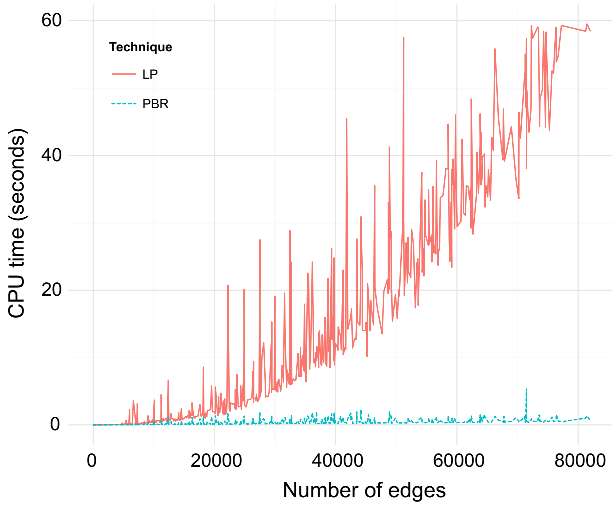

Fig. 3. CPU time for LP relaxation and incremental PBR computation.

Lodi, 2007). Obtaining a good incumbent solution at an early stage of the B&B process can both improve on the upper bound and accelerate the search due to better fathoming. In particular, the fathoming operation discards a subset of the solution space if the lower bound at a search node is greater than the incumbent solution. The incremental PBR computation time is small and does not increase with problem size, the PBR solution quality exceeds that of LP relaxation in all tests, and is demonstrably better than the default collection of heuristics, pre-processing algorithms, and other techniques built into modern solver software for the larger problem instances. *Improvement* heuristics are used to improve the incumbent solution at branching nodes, and examples include local branching (Fischetti & Lodi, 2003) and the relaxation induced neighborhood search (Danna, Rothberg, & Le Pape, 2005). Since the LP relaxation is updated at each B&B node, the probabilities from the logistic regression predictive model can be updated at each node as well. However, during initial testing, we find PBR to be best employed as a primal heuristic. As such PBR will be incorporated as a primal heuristic and evaluate solution quality for a variety of network instances against that of commercial MIP solver.

The test set includes a total of 30 randomly generated instances of varying sizes; ten problems have 100 nodes, ten have 300 nodes, and the largest set of ten instances are comprised of 900 nodes. Each instance is generated as described in Section 3.1 and has a high fixed to variable cost ratio. The number of edges range from 394 to 72,276. The objective of this experiment is to evaluate the performance of PBR as primal heuristic in searching for an exact solution. However, practically speaking many hard problems are not solved to optimality. In this experiment, a maximum run time of 3600 s for each test is permitted. Some of the instances cannot be solved in only one hour. The best objective found using the PBR primal heuristic with the commercial solver as well as the best objective found using the default solver strategy alone within the time limit are recorded as $z_{\text{PBR-primal}}$ and $z_{\text{GRB-default}}$, respectively.

Let z_{gap} denote the percentage gap between $z_{\text{PBR-primal}}$ and $z_{\text{GRB-default}}$ be defined as

$$z_{\text{gap}} = \frac{z_{\text{PBR-primal}} - z_{\text{GRB-default}}}{|z_{\text{GRB-default}}|} \times 100\%.$$

The results are reported in Table 4. The p-value results of the paired t-test on the instances with 100 and 300 nodes are 0.33 and 0.78, respectively. Statistically, there is no difference in the solution quality results for these two problem classes. However, for the largest problem class with 900 nodes, the p-value for the paired t-test is 0.004. The average z_{gap} is -4.07%. The default commercial solver alone only outperformed the PBR hybrid approach in one of ten hard problems, whereas the new hybrid technique outperformed the default software in eight out of ten hard problems. In three of these cases, the improvements exceed 8% points.

Table 4
Objective values and gaps report.

Nodes	Arcs	$z_{\text{PBR-primal}}$	$z_{\text{GRB-default}}$	z_{gap} (%)
100	494	2061558.5	2061558.5	0.00
	1044	1979088.4	**1963950.0**	0.77
	706	1992384.1	1992384.1	0.00
	630	1875760.1	1875760.1	0.00
	640	1817097.8	1817097.8	0.00
	530	1858583.9	1858583.9	0.00
	394	**2523919.8**	2526065.6	−0.08
	654	1781751.0	**1778756.3**	0.17
	1034	1302851.6	1302851.6	0.00
	600	1997502.1	1997502.1	0.00
300	3320	**5007261.7**	5043679.9	−0.72
	8376	3609142.9	3609142.9	0.00
	5262	4673005.4	4673005.4	0.00
	4568	4496875.7	**4447349.7**	1.11
	4660	**4745624.5**	4755089.9	−0.20
	3650	5238362.1	5238362.1	0.00
	2404	5520568.4	5520568.4	0.00
	4786	4530403.6	**4528409.5**	0.04
	8286	**4284231.6**	4310106.4	−0.60
	4304	4823258.2	4823258.2	0.00
900	26464	**14446116.2**	14880751.9	−2.92
	72276	**11818928.8**	12094918.3	−2.28
	44070	**13171786.0**	14574145.8	−9.62
	37784	**13726017.7**	15042070.8	−8.75
	38618	**13491715.7**	13790571.2	−2.17
	29458	**15662131.2**	17061309.4	−8.20
	18178	14463571.1	14463571.1	0.00
	39742	**13518757.4**	14540684.5	−7.03
	71468	12305509.0	**12266906.2**	0.31
	35390	**13488456.0**	14346083.8	−5.98

Superior objective values are highlighted in bold.

4. Conclusions

This paper introduces an original and novel heuristic technique for the FCNF problem. To the best of our knowledge, PBR is the first method that inherently employs information from a statistical learning model to solve a mixed integer programming problem. PBR replaces the original edge flow cost with a function of predictive model probabilities. In this implementation, the model proposed in Nicholson and Zhang (2016) is used as the underlying probabilistic classifier and proves to be a valid model. Empirical analysis validates the efficiency and effectiveness for PBR as an independent heuristic and as a primal heuristic to improve an exact search. For larger problems, the incremental computational time is negligible. PBR consistently improves on the FCNF LP relaxation solution across all problem difficulty levels evaluated. Additionally, compared to the variety of algorithms and heuristics currently incorporated in commercial solvers, PBR provides a favorable alternative (assuming the predictive model underpinning the technique is of sufficient accuracy and efficiency). These characteristics support the concept of integrating PBR as a primal heuristic for FCNF problems. Empirically, this strategy produces a statistically significant improvement in the solution quality for large FCNF instances.

This work demonstrates some of the potential of incorporating statistical learning techniques with traditional optimization techniques. For the PBR implementation to work well, the underlying model should be accurate and the associated predictive probabilities should be computational easy to obtain. The FCNF is a classical NP-hard problem with wide application. The nature of the FCNF allows for certain consistent descriptors (e.g., fixed cost to variable cost ratio, average supply per node, etc.) which can be used to develop predictive models. While logistic regression forms the base of the implementation in this paper, any probabilistic classifier can be used. Supervised learning methods for model building are dependent on the available data and the particular characteristics appropriate to various optimization problems. If more specific problem features are relevant (e.g., edge capacities in a capacitated networks, granular commodity requirements for a multiple commodity problem, etc.) or another supervised learning approach is more accurate, a different predictive model may be developed. The empirical results in this study, based on a relatively simple predictive model applied to a difficult optimization problem, are encouraging.

References

Achterberg, T., & Berthold, T. (2007). Improving the feasibility pump. *Discrete Optimization, 4*(1), 77–86.

Adlakha, V., & Kowalski, K. (2010). A heuristic algorithm for the fixed charge problem. *Opsearch, 47*(2), 166–175.

Antony Arokia Durai Raj, K., & Rajendran, C. (2012). A genetic algorithm for solving the fixed-charge transportation model: Two-stage problem. *Computers & Operations Research, 39*(9), 2016–2032.

Armacost, C., Barnhart, A., & Ware, K. (2002). Composite variable formulations for express shipment service network design. *Transportation Science, 36*(1), 1–20.

Balinski, M. (1961). Fixed-cost transportation problems. *Naval Research Logistics Quarterly, 8*(1), 41–54.

Barr, R., Glover, F., & Klingman, D. (1981). A new optimization method for large scale fixed charge transportation problems. *Operations Research, 29*(3), 448–463.

Bertacco, L., Fischetti, M., & Lodi, A. (2007). A feasibility pump heuristic for general mixed-integer problems. *Discrete Optimization, 4*(1), 63–76.

Berthold, T. (2007). RENS-relaxation enforced neighborhood search. *Konrad-Zuse-Zentrum für Informationstechnik*.

Bixby, M., Fenelon, E., Gu, Z., Rothberg, E., & Wunderling, R. (2000). *MIP: Theory and practice closing the gap. System modelling and optimization* : . Springer, pp. 19–49.

Cabot, A., & Erenguc, S. (1984). Some branch-and-bound procedures for fixed-cost transportation problems. *Naval Research Logistics Quarterly, 31*(1), 145–154.

Costa, A. (2005). A survey on benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research, 32*(6), 1429–1450.

Danna, E., Rothberg, E., & Le Pape, C. (2005). Exploring relaxation induced neighborhoods to improve mip solutions. *Mathematical Programming, 102*(1), 71–90.

Driebeek, N. (1966). An algorithm for the solution of mixed integer programming problems. *Management Science, 12*(7), 576–587.

El-Sherbiny, M., & Alhamali, R. (2013). A hybrid particle swarm algorithm with artificial immune learning for solving the fixed charge transportation problem. *Computers & Industrial Engineering, 64*(2), 610–620.

Fischetti, M., & Lodi, A. (2003). Local branching. *Mathematical Programming, 98*(1–3), 23–47.

Fügenschuh, A., & Martin, A. (2005). Computational integer programming and cutting planes. *Handbooks in Operations Research and Management Science, 12*, 69–121.

Ghamlouche, I., Crainic, T., & Gendreau, M. (2003). Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Operations Research, 51*(4), 655–667.

Goemans, Michel X., & Williamson, David P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM), 42*(6), 1115–1145.

Hewitt, M., Nemhauser, G., & Savelsbergh, M. (2010). Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem. *INFORMS Journal on Computing, 22*(2), 314–325.

Jarvis, J., Rardin, R., Unger, V., Moore, R., & Schimpeler, C. (1978). Optimal design of regional wastewater systems: A fixed-charge network flow model. *Operations Research, 26*(4), 538–550.

Kennington, J., & Unger, E. (1976). A new branch-and-bound algorithm for the fixed-charge transportation problem. *Management Science, 22*(10), 1116–1126.

Kim, D., & Pardalos, P. (1999). A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. *Operations Research Letters, 24*, 195–203.

Lederer, P., & Nambimadom, R. (1998). Airline network design. *Operations Research, 46*(6), 785–804.

Mahajan, A. (2010). Presolving mixed-integer linear programs. *Wiley Encyclopedia of Operations Research and Management Science*.

Melo, M. Teresa, Nickel, Stefan, Saldanha-da Gama, Francisco, et al. (2009). Facility location and supply chain management – A review. *European Journal of Operational Research, 196*(2), 401–412.

Molla-Alizadeh-Zavardehi, S., Hajiaghaei-Keshteli, M., & Tavakkoli-Moghaddam, R. (2011). Solving a capacitated fixed-charge transportation problem by artificial immune and genetic algorithms with a prüfer number representation. *Expert Systems with Applications, 38*(8), 10462–10474.

Monteiro, M., Fontes, D., & Fontes, F. (2011). An ant colony optimization algorithm to solve the minimum cost network flow problem with concave cost functions. *GECCO*, 139–146.

Nemhauser, G., & Wolsey, L. (1988). *Integer and combinatorial optimization* (Vol. 18). Wiley New York.

Nicholson, C., & Zhang, W. (2016). Optimal flow analysis, prediction and application. <http://oklahomaanalytics.com/wp-content/uploads/2016/05/OFA-PrePrint.pdf>.

Nozick, L. (2001). The fixed charge facility location problem with coverage restrictions. *Transportation Research Part E: Logistics and Transportation Review, 37*(4), 281–296.

Optimization, Gurobi (2012). Inc. Gurobi optimizer reference manual, version 5.0.

Ortega, F., & Wolsey, L. (2003). A branch-and-cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow problem. *Networks, 41*(3), 143–158.

Palekar, U., Karwan, M., & Zionts, S. (1990). A branch-and-bound method for the fixed charge transportation problem. *Management Science, 36*(9), 1092–1105.

Steinberg, E., & Napier, H. (1980). Optimal multi-level lot sizing for requirements planning systems. *Management Science, 26*(12), 1258–1271.

Sun, M., Aronson, J., McKeown, P., & Drinka, D. (1998). A tabu search heuristic procedure for the fixed charge transportation problem. *European Journal of Operational Research, 106*(2), 441–456.

Wolsey, L. (1998). *Integer programming* (Vol. 42) New York: Wiley.