Working Paper Series

HEARIN CENTER FOR ENTERPRISE SCIENCE

HCES-01-04

Solving Fixed Charge Capacitated Multicommodity Network Design Problems Using Guided Design Search

By

Mark W Lewis

The University of Mississippi

Director, Keith Womer
School of Business Administration
The University of Mississippi
Post Office Box 1848
University, MS 38677-1848
(662) 915-5820
http://hces.bus.olemiss.edu

Solving Fixed Charge Capacitated Multicommodity Network Design Problems Using Guided Design Search

Mark Lewis[a,†],

a Hearin Center for Enterprise Science, School of Business, University of Mississippi, University, MS 38677, USA. mlewis@bus.olemiss.edu

Latest Revision: January , 2004.

Abstract —In this paper we introduce a preprocessing technique using experimental design sampling to generate the estimated effects of binary decision variables on the objective function. The binary mixed-integer programs are recast with elastic constraints during the sampling to ensure feasibility. The effects estimates are useful for ranking the decision variables and can also be used to reduce and refine the search of the problem space. By setting a small number of significant variables to their appropriate levels, we find better solutions faster than the industry standard for the difficult fixed charge capacitated multicommodity network design problem. Our results reinforce the translation of the concept of sparsity of variable effects from the experimental design domain to integer programs, that is, generally a small number of variables will have the most significant effects on solution quality and time.

Keywords: Network design, Elastic constraints, Experimental design.

1. Introduction

† This research was supported by The University of Mississippi School of Business Administration and the Office of Research and Sponsored Programs.

In this paper we describe the use of efficient experimental design sampling techniques to calculate unbiased estimates of the binary variables' average effect on the objective function of a mixed binary integer program. To ensure feasibility during the sampling (preprocessing) phase the problems are recast with elastic constraints (see Chinneck and Dravnieks 1991). This use of sampling is an extension of the ideas presented in Karger (1999) where proofs for using random sampling to reduce and solve, to a high probability, a wide variety of cut-dependent undirected graph problems are presented. Multi-dimensional testing with large instances of the fixed-charge capacitated multi-commodity network design problem shows that setting a small percentage of significant variables to their estimated optimal values generally creates better solutions that are obtained faster than the industry standard optimization software CPLEX (using version 8.1 default settings).

Finding the average effect of each binary variable on the objective function is a form of knowledge discovery, i.e. we are mining data to reveal problem structure. The discovery of average variable effects is useful for ranking the decision variables by their effect and can be used to guide a solution search. Guiding is accomplished by providing priorities for variable selection and branching direction within a search framework such as branch-and-bound (see Linderoth and Savelsbergh 1999 for a survey of search strategies within a branch-and-bound framework.). Our approach, named Guided Design Search (GDS), can be used stand-alone or integrated with an exact or heuristic search. In this paper, we apply GDS heuristically by setting the most significant binary variables to their appropriate levels, followed by default CPLEX processing. We find that the solution quality and time are greatly influenced by setting a certain, small fraction (generally $< 3\%$) of the total number of binary variables.

The fixed-charge capacitated multi-commodity network design problem (FCMD) minimizes arc fixed and path flow costs subject to capacity and multicommodity demand constraints (see Ahuja et al. 1993). Fixed charge network design problems find application in many different areas, including telecommunication network design, highway design, electrical power grids, water supply, network topology design and network capacity expansion (see Magnanti and Wong 1984 and Luss 1982). The fixed-charge capacitated multi-commodity network design problem encompasses the multi-commodity network flow problem, that is, given the arcs to be included in the network and their capacity, FCMD reduces to a capacitated multi-commodity network flow problem (which is still a difficult problem to solve, see surveys by Assad 1978 and Kennington 1978).

The algorithms described by Karger (1999) are based on constructing graph skeletons, i.e. the same set of vertices but with a smaller random sample of edges. He shows that these skeleton graphs preserve the cut information, but are much smaller and easier to solve than the graphs they represent. He also shows applicability to various types of network design problems, including the fixed-charge network design problem in both uncapacitated and capacitated forms.

The uncapacitated equivalent of FCMD eliminates the capacity restrictions on the arcs and is an adequate model in cases where the capacities are very large relative to the maximum flows to be encountered. Past approaches to solving the uncapacitated fixed-charge multi-commodity network design problem have included Lagrangian relaxation within branch-and-bound, (Holmberg 1998), benders decomposition (Magnanti and Wong 1986), dual-ascent procedures (Balakrishnan et al. 1989) and capacity improvement (Lamar et al 1990). All these specialized approaches have shown better lower bounds or generally faster solution times than branch-and-bound using LP relaxations.

The capacitated form of the network design problem, FCMD, is more generally applicable, but difficult to solve. Herrmann et al (1994) extended the uncapacitated network design problem work of Balakrishnan et al (1989) to FCMD. Their dual-ascent approach generates better lower bounds than the linear programming relaxation and their increasing arc cost method generates good feasible solutions. Crainic et al (2000) present a simplex-based tabu search procedure that finds good upper bounds (feasible solutions) to FCMD. Their tabu search method performed better than CPLEX default branch-and-bound, a greedy heuristic, and a resource-decomposition heuristic. They note that the number of commodities significantly influences the difficulty of their problems.

Holmberg and Yuan (2000) extend to the capacitated problem the Lagrangian branch-and-bound that they used for the uncapacitated network design problem (1998). Their method can also be implemented as either exact or heuristic. The heuristic method is based on the frequency that an arc is included (or not included) in the solution to the subgradient search. The arc decision variable is set to 1 (0) for arcs included at a higher (lower) frequency. This heuristic approach is similar to the one to be described in this paper. Fixing variables in the branch-and-bound tree reduces the search space so that good, feasible solutions are found quickly and optimal solutions are (hopefully) not cut out of the search.

The contribution of this paper is its description and implementation of heuristic Guided Design Search, consisting of recasting the problem with elastic constraints, experimental design sampling, and rule generation. We present the results of applying GDS to heuristically solve an NP-hard problem within the established framework of CPLEX's callable subroutines. In addition, we show that a relatively small number of significant variables have a dramatic effect on solution quality and time.

This paper is not comparing CPLEX's preprocessing of mixed-integer problems to experimental design sampling (see Kennington and Lewis for a network preprocessing comparison). GDS provides complementary information, for integration with other preprocessing techniques such as bound strengthening, coefficient reduction, and variable substitution (see ILOG Optimization Suite documentation).

This paper is organized as follows. Section 2 presents the mathematical model of the fixed charge capacitated multicommodity network design problem. Section 3 describes basic experimental design concepts and how they relate to binary integer programs. The need for elasticizing the constraints and why the three sigma rule was implemented for testing is also presented in Section 3. The results of the multidimensional testing are reported in Section 4. Conclusions and further research opportunities are offered in Section 5.

2. Mathematical Model

The mathematical model presented is consistent with those found in the literature that use an arc-path formulation of network flow (see Crainic et al. 2000, Doucette et al. 1999, 2001). The model is modified in the next section to include elastic constraints (Chinneck and Dravnieks 1991, Holder 2000) during the pre-processing phase. Modeling with elastic constraints penalizes the use of slack and surplus variables for satisfying certain constraints and guarantees feasibility when setting the binary variables.
We have chosen the arc-path formulation over a node-balance form (Kennington and Lewis 2001), because in network routing problems it can be desirable to have complete control over which paths are allowed into the solution set (see Dunn et al. 1994, Iraschko et al. 1998, 2000). The node balance form for network flows may provide

lower cost solutions, but long paths may be formed, and long paths may not be acceptable in practice (Doucette et al. 2001). Another disadvantage of the node balance form is that the size of the problem (i.e. number of variables) is largely determined by the number of nodes and arcs, while in the arc-path form the size of the problem can be controlled by restricting the number of enumerated paths without reducing the topology.

Let $[N,E]$ be a network with a set of nodes N and a set of undirected arcs E. Undirected arcs are used because the telecommunication network applications driving this research are best modeled with undirected arcs. $E = \{e_1, \ldots, e_i, , \ldots, e_{\overline{m}}\}$ is the set of ordered pairs of nodes, that is, $e_i = (n, m)$ with $n \neq m$ and $n, m \in N$, so that the binary variable y_i denotes whether or not arc e_i is to be included in the network design. The fixed cost to add an arc e_i to the network is $\;_i$. Let $D = \{(i, j) \in N : d_{ij} > 0\}$ where d_{ij} denotes the amount of demand (bandwidth needed) between nodes i and j.

Let P denote the index set of enumerated paths in $[N,E]$ between nodes $(i, j) \in D$, where a *path* in $[N,E]$ is defined as an ordered sequence of nodes, $p_k = \{n_i , \ldots, n_j\}$. Thus, $P = \{k: p_k$ is a path in $[N,E]$ $\}$. In order to use the term origin-destination pair we define the *origin* of p_k as node n_i and the *destination* of p_k is node n_j. The value of $p_k \in P$ is the amount of flow between its origin and destination nodes.

$C = \{ c_i : i \in E \}$ is the set of upper bound capacity limits for each arc (or edge) in E. In our testing, c_i is the amount of capacity commonly associated with the fiber optic links in a telecommunications network and is set equal to 48. If an arc is included in the design, it is at its' capacity limit (for capacity expansion problems with variable capacity limits see Dahl and Stoer 1998 and Kennington and Lewis 2001). $B = \{ b_k : k \in P\}$ is the set of costs per unit flow on path k, where b_k is the sum of the unit flow costs of the individual arcs comprising path k. Note that this model does not require the path cost to be the sum of its component arc flow costs. This is useful in modeling quality of service demand routing in telecommunication networks where short paths may cost more than longer paths (Ramaswami 2002).

The remaining sets needed for the P_{FCMD1} problem formulation are defined as follows. The set of all paths between an origin i and destination j is defined as: $O_{ij} = \{k : \text{origin}(p_k) = i, \text{destination}(p_k) = j, k \in P\}$. Let $F^e = \{k : e \in p_k, k \in P \}$ denote the index set of paths which contain arc e.

P_{FCMD1} $(N, E \quad D, P, C, \quad , B) = \{$

$$\min \sum_{\forall i \in E} \;_i\, y_i + \sum_{\forall k \in P} b_k\, p_k \qquad (1)$$

s.t.

$$\sum_{k \in F^i} p_k \leq c_i\, y_i , \qquad \forall i \in E \qquad (2)$$

$$\sum_{k \in O_{ij}} p_k \geq d_{ij} , \qquad \forall (i, j) \in D \qquad (3)$$

$$y_i \in \{0,1\} \qquad \forall i \in E \qquad (4)$$

$p_k \geq 0 \qquad \forall k \in P \qquad (5)$

}

The objective function (1) minimizes the fixed arc and path flow costs subject to meeting demand and capacity constraints. Capacity constraints (2) require that the total amount of flow on an arc *i* from all paths using that arc be less than the amount of capacity implemented. Demand inequalities (3) set the minimum amount of bandwidth required between an origin-destination pair.

Consistent with the standard form of FCMD, the path variables (5) are not integer. In some telecommunications modeling, the p_k variables may be required to be integral, or, may require that only 1 path be used for all flow. In this investigation we stay closer to the more general network design model and save these more specific models for future research.

3. Experimental Design Testing to Preprocess the Solution Space

Taguchi methods (1989) stress quality control early in the design process in order to enhance quality during production operations. A key element of Taguchi methods is the use of experimental designs, a.k.a. design of experiments (DOE). In short, DOE is used to help determine the critical factors affecting a given performance parameter, e.g. what temperature and pressure settings optimize product yield? Benefits of DOE testing include unbiased sampling (versus random sampling or simply averaging observations made during branch-and-bound or a metaheuristic search) and interaction effects estimation (DeVor et al. 1992). *Main effect estimation* is the estimated average effect of a variable on the output parameter (in our case, the objective function value). When two variables interact to create an effect that is different from the sum of their individual effects, this interaction effect can also be estimated.

A test run in a DOE corresponds to the output value measured after setting the variables at the levels determined by the experimental design. The number of test runs, *k*, needed for a 2-levels-per-factor-*k* DOE grows at the rate of 2^k. Two-level DOEs are popular because they are easier to design and interpret than higher level designs and, in our case, they correspond to binary variables in integer programs. Alternatives to 2-level DOE are Latin-Square designs, random sampling or testing by changing one-factor-at-a-time (Berger and Mauer 2002).

In a production environment test runs are generally expensive. However, GDS test runs are generally not expensive, with each test run sample usually being solved in fractions of a second. Although testing is fast, the number of test runs per factor grows at an exponential rate, therefore confounding techniques (also known as fractional experimental designs) are employed to reduce the number of tests. Variables that are confounded in a fractional DOE have their own effects as well as others, combined (i.e. confounded) into one estimate (DeVor et al. 1992). Fortunately, sequential testing (e.g. using the "mirror" of the original experiment) can remove the effects of confounding and reveal the main effects (Mason et al. 1989). Because sequential mirror testing is based on the assumption that interactions between three variables (or more) are insignificant, the justification for ignoring high-order interactions follows.

The concept of "*sparsity of variable effects*" (see Box and Meyer, 1986) states that, in general, when many variables are examined for their effect on a performance parameter, only a relatively few variables will have a major effect. For example, "sparsity of variable effects" says that the 5-factor interaction $x_1 x_2 x_3 x_4 x_5$ probably has at least one factor with a very small or 0 value, thus making the 5-factor interaction very small or 0. In a large DOE there are a large number of these multiple, redundant interactions, but most are small and can be ignored. This is similar to the Pareto Principal -- 80% of effects are due to 20% of the variables.

While it is not uncommon to make the assumption that interactions between 3 or more variables will have negligible effects and can therefore be ignored, it is to be expected that the optimization of combinatoric problems is difficult because there are high order interaction effects. Thus, part of the research in this paper is to discover the applicability of these standard assumptions when applied to mixed-binary integer problems. In other words, are there really only a relatively small number of variables that have significant effects in large binary programs and is it acceptable to ignore higher-order interaction effects?

3.1 Elastic Constraints

Given a vector **Y** of decision variables y_i, P_{FCMD1} becomes a minimum cost multi-commodity flow problem, which may or may not be feasible. To account for infeasibilities when provided a **Y** from the table of DOE test runs, the problem form P_{FCMD2} with elastic constraints is used during preprocessing. The penalty term M affects the quantification of infeasibility, we used M = 10000 for the results presented here.

P_{FCMD2} $(N, E \quad D, P, C, \qquad B, M) = \{$

$$\min \sum_{\forall i \in E} {}_i\, y_i + \sum_{\forall k \in P} b_k\, p_k + M \sum_{\forall i \in E} S_i + M \sum_{\forall (i,j) \in D} S_{ij} \tag{6}$$

s.t.

$$\sum_{k \in F^i} p_k - S_i \leq c_i\, y_i\,, \qquad \forall i \in E \tag{7}$$

$$\sum_{k \in O_{ij}} p_k + S_{ij} \geq d_{ij}\,, \qquad \forall (i, j) \in D \tag{8}$$

$$y_i \in \{0,1\}\,, \qquad \forall i \in E \tag{9}$$

$$p_k \geq 0 \qquad \forall k \in P \tag{10}$$

$$S_{ij} \geq 0, \qquad \forall (i, j) \in D \tag{11}$$

$$S_i \geq 0 \qquad \forall i \in E \tag{12}$$

}

3.2 Rules

There are several ways of utilizing GDS, for example, the results can be integrated into either an exact or heuristic search. In the exact approach GDS provides guidance in the form of branching directions and variable selection, while in the heuristic approach certain binary variables are set to the appropriate level discovered during testing. The best rule(s) to determine which variables to set at what level is an open question whose answer is most likely tied to the structure of the problem (including interactions).

While using a single rule for all problem types may not be the best approach, this paper tests the results of implementing a single rule for setting a small number of significant variables. It is anticipated that the reduced solution space will then yield time improvements without sacrificing solution quality. After preliminary tests with various problem types (mostly multi-commodity flow, fixed-charge network, robust network and capacity allocation problems), we found that most variable effect distributions have a near Gaussian normal appearance with identifiable potential outliers, i.e. variables with consistently significant effects. Based on this, the rule implemented for testing was to set all variables that were three standard deviations from either side of the mean of the sample. Figure 1 shows a representative frequency distribution of variable objective function effects and illustrates the normal appearance of the distribution as well as potential outliers beyond three sigma from the mean. Future research can investigate the effects of and interactions between various rules and parameters and algorithm effectiveness (see Barr et al. 1995 and Rardin and Reha 2001 on testing methods).

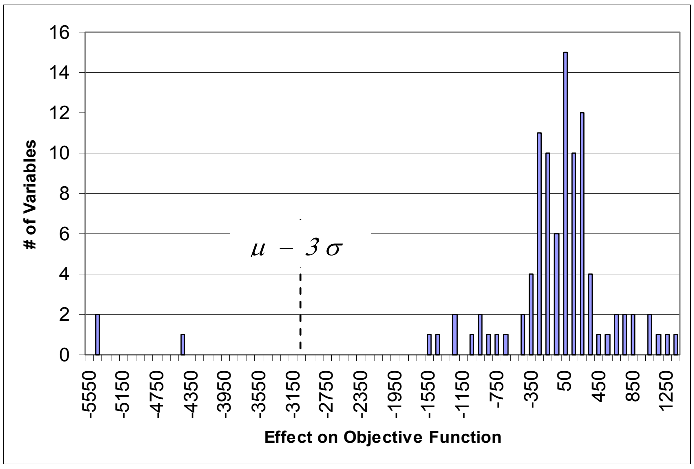

Figure 1. Example result from DOE testing showing outliers

4. Results of Testing

The main objectives of the testing are to compare the performance between GDS/CPLEX and CPLEX and to determine the effect of problem characteristics such as graph connectivity, number of commodities, number of nodes, and arc capacity. The problems generated all have parallel arcs (allowing bi-directional flow between nodes) and one commodity of demand between origin-destination pairs. The path set generated consisted of the k-shortest paths (Dunn et al. 1994) plus a set of hop-limited paths. The four parameters used to create the problem instances are the number of nodes, the connectedness of the graph, the arc capacity, and the number of commodities. The levels for each parameter are shown in Table 1. The tests were run on a Dell PowerEdge server with quad 2.4 GHz processors (no parallel processing).

Network Parameter	Possible Settings
# Nodes	10 / 20 / 30 / 40 / 50
# Arcs	Fully connected network / 50% connected
Arc Capacity	12 / 48
Commodities	Full / 50% assigned

Table 1. Problem Test Set Characteristics

While in most cases, the solution benefited from the pre-processing, in some a large amount of time was spent for little advantage over default CPLEX. On average, GDS preprocessing time was ~30% of the total processing time. Thus, GDS preprocessing may best be an option for problems that are known to be difficult. Table 2 summarizes the preprocessing times and number of variables set in the 40 problems and illustrates that on average, <3% of the total number of binaries were set (half of the problems have < 1.5% set). Table 2 supports the idea that a relatively small number of variables play large parts.

Tables 3 through 6 categorize the test results by the number of candidate arcs and number of demands. This dimensional analysis provides insight into the GDS/CPLEX comparison. The tables show the best solution found by GDS and default CPLEX and the times to that solution within the 24 hour processing limit networks of various size and capacity. The GDS times are the sum of the GDS preprocessing time plus the time in CPLEX branch-and-cut. Because the two approaches often did not discover the same solution during their searches, the time difference is not the difference to the same solution. Therefore, when reviewing the table, note both the solution quality and the time to the solution. Problem ID #19 with a high number of demands and fewer candidate arcs did not have enough arc capacity to allow a feasible solution, but is included in the table for completeness.

Table 3 illustrates that for high-demand networks with a high number of candidate arcs, GDS was faster overall to the same or better solution. Table 4 illustrates that for networks with fewer numbers of candidate arcs and a high number of demands, GDS performance varied from being faster to the same solution (3 problems), slower but to a better solution (2 problems), faster to a worse solution (1 problem) and about the same performance. Table 6 illustrates that for networks with a high number of candidate arcs but fewer demands, GDS was slower than CPLEX on 6 problems. This is mainly because of the time spent in preprocessing these relatively easy problems.

Table 5 reveals that GDS consistently found the same or better solutions faster than CPLEX on networks having 50% of total possible demands and arcs. Although these may be viewed as the “easiest” problems, they were still quite difficult, with only the two smallest problems solved to optimality and an average optimality gap of 33% after 24 hours of processing for the remaining problems. Careful testing shows that GDS performance is affected by problem characteristics. This raises the question for future research whether the rule for setting variables should use not only a statistical analysis, but also take into account problem characteristics.

ID #	# Binary Variables	# Binaries Set	% Of Binaries	GDS Preprocessing Time (Sec)	% Of Total Time To Best Solution
1	45	4	8.9%	6	4%
2	45	5	11.1%	6	17%
3	190	6	3.2%	50	11%
4	190	4	2.1%	44	31%
5	435	7	1.6%	220	61%
6	435	7	1.6%	172	14%
7	780	8	1.0%	936	15%
8	780	10	1.3%	940	48%
9	1225	7	1.3%	1,440	94%
10	1225	15	1.2%	456	5%
11	23	2	8.9%	1	33%
12	23	2	8.9%	1	33%
13	100	4	4.0%	41	10%
14	100	2	2.0%	29	8%
15	225	7	0.0%	225	86%
16	225	7	3.1%	158	0.3%
17	400	1	2.3%	392	6%
18	400	6	1.5%	99	21%
19	500	---	----	---	---
20	500	1	0.8%	133	0.6%
21	23	2	8.9%	0.4	29%
22	23	2	8.9%	0.5	20%
23	100	3	3.0%	8	44%
24	100	2	3.0%	8	3%
25	225	4	1.8%	25	0.1%
26	225	1	1.8%	25	42%
27	400	4	1.0%	87	4%
28	400	4	1.0%	87	1%
29	500	2	1.2%	176	0.5%
30	500	6	1.2%	171	93%
31	45	2	4.4%	3	50%
32	45	2	4.4%	3	75%
33	190	2	1.1%	4	12%
34	190	2	1.1%	4	46%
35	435	1	0.5%	16.9	83%
36	435	4	0.9%	14.9	4%
37	780	3	0.4%	199	93%
38	780	4	0.5%	175	61%
39	1225	3	0.3%	389	3%
40	1225	14	1.1%	303	2%
	Averages	4	2.8%	181	30%

Table 2. Summary for Number of Binary Variables Set & GDS Preprocessing Time

ID #	N	Pos. Arcs	Arc Cap	D	CPLEX Time to best sol.	GDS Total time to best	Time Dif.	CPLEX Best sol.	GDS Best sol.	Sol. Dif.
1	10	45	12	45	150	142	-8	230	230	0
2	10	45	48	45	30	36	6	180	180	0
3	20	190	12	190	1577	444	-1133	980	970	-10
4	20	190	48	190	100	144	44	570	570	0
5	30	435	12	435	1001	360	-641	2394	2394	0
6	30	435	48	435	31422	1272	-30150	3510	3510	0
7	40	780	12	780	23800	6336	-17464	4476	4476	0
8	40	780	48	780	1000	1940	940	2320	2320	0
9	50	1225	12	1225	220	1525	1305	9110	9110	0
10	50	1225	48	1225	42900	8456	-34444	8786	8786	0
						Totals	-81547			-10

Table 3. GDS/CPLEX Comparison on Networks with a High Number of Demands and High Number of Candidate arcs

ID #	N	A	Cap	D	CPLEX Time to best sol.	GDS Total time to best	Time Dif.	CPLEX Best sol.	GDS Best sol.	Sol. Dif.
11	10	23	12	45	2	3	1	270	270	0
12	10	23	48	45	2	3	1	522	522	0
13	20	100	12	190	583	416	-167	1052	1052	0
14	20	100	48	190	16840	353	-16487	1532	1532	0
15	30	225	12	435	78	261	183	2394	2394	0
16	30	225	48	435	9426	49701	40275	3510	3462	-48
17	40	400	12	780	34607	14313	-20294	4284	4284	0
18	40	400	48	780	329	481	152	8328	7752	-576
19	50	500	12	1225	---	----	----	---	----	----
20	50	500	48	1225	7200	1000	-6067	11762	12050	288
						Totals	-2403			-336

Table 4. GDS/CPLEX Comparison on Networks with a High Number of Demands and Low Number of Candidate arcs

ID #	N	A	Cap	D	CPLEX Time to best sol.	GDS Total time to best	Time Dif.	CPLEX Best sol.	GDS Best sol.	Sol. Dif.
21	10	23	12	20	2	1.4	-0.6	160	160	0
22	10	23	48	20	3	2.5	-0.5	472	472	0
23	20	100	12	50	2104	18	-2086	400	388	-12
24	20	100	48	50	1110	311	-799	1012	1012	0
25	30	225	12	100	200	35	-165	752	740	-12
26	30	225	48	100	22011	60	-21951	2024	2024	0
27	40	400	12	200	16431	2442	-13989	1480	1456	-24
28	40	400	48	200	195	6537	6342	4240	4192	-48
29	50	500	12	400	42329	22650	-19503	1768	1756	-12
30	50	500	48	400	200	183	-17	5824	5776	-48
						Totals	-52503			-156

Table 5. GDS/CPLEX Comparison on Networks with a Low Number of Demands and Low Number of Candidate arcs

ID #	N	A	Cap	D	CPLEX Time to best sol.	GDS Total time to best	Time Diff.	CPLEX Best sol.	GDS Best sol.	Sol. Diff.
31	10	45	12	20	3	6	3	160	160	0
32	10	45	48	20	1	4	3	472	472	0
33	20	190	12	50	15	31	16	376	376	0
34	20	190	48	50	5	9	4	1012	1012	0
35	30	435	12	100	7	31	24	668	668	0
36	30	435	48	100	1814	365	-1449	1640	1640	0
37	40	780	12	200	85	213	128	1384	1384	0
38	40	780	48	200	1238	285	-953	2464	2464	0
39	50	1225	12	400	1946	14874	12928	2408	2408	0
40	50	1225	48	400	55733	15847	-39886	3920	3872	-48
						Totals	-29182			-48

Table 6. GDS/CPLEX Comparison on Networks with a Low Number of Demands and High Number of Candidate arcs

Problem 20 was the only instance where GDS did not find as good (or better) a solution as CPLEX in the allotted 24 hours. Investigating this apparent anomaly, we set only the one variable with the largest effect and re-ran the test, and even though this variable formed part of the CPLEX solution, similar results were obtained. The exact reason that providing a part of the answer resulted in poorer performance is unclear, but may be related to the preprocessing and/or search method utilized (automatically determined by CPLEX). To further investigate if GDS was providing incorrect variable settings, we set 50% of the largest variables that were more than one standard deviation from the average (total of 32 variables). In this case, GDS found in 100 seconds the solution CPLEX obtained in 7200 seconds. Thus, GDS does seem to be providing the correct guidance. In general, setting a larger number of variables did not produce better solutions for these problems, which may be due to statistical error in the estimates, penalty size for the elastic constraints, or alternate optima with exclusivity characteristics.

Table 7 summarizes the performances by network parameter, that is, the data is organized by network parameter and averages calculated. For example, we see that, *on average* there is not much time or solution quality difference between CPLEX and GDS for the 10 node problems, but for the 40 node problems, GDS was faster to a better average solution. For the 50 node problems, CPLEX was slower to a better average solution. GDS performed better on the 50% connected candidate topologies than on the 100% connected. The average performance improvement of GDS over CPLEX is not dramatically affected by the number of demands. The time and solution difference average over all tests was -4238 seconds and -14 capacity units, thus, on average, GDS was faster to better solutions.

Network Parameter Setting	CPLEX Time to best sol.	GDS Total time to best sol.	Average Time Difference	CPLEX Best Solution	GDS Best Solution	Average Solution Difference
10 nodes	24	25	1	308	308	0
20 nodes	2792	178	-2614	867	864	-3
30 nodes	8245	6506	-1739	2112	2104	-8
40 nodes	9711	4068	-5642	3622	3541	-81
50 nodes	21504	9263	-12241	6225	6251	23
Fully connected	8152	2616	-5536	2353	2350	-3
50% connected	8087	5197	-2890	2726	2700	-25
Capacity 12	6586	3383	-3203	1829	1825	-4
Capacity 48	9578	4339	-5239	3205	3181	-24
Full demands	9014	4596	-4418	3485	3467	-17
50% demands	7272	3187	-4084	1632	1622	-10

Table 7. Average Values by Network Parameter

5. Conclusions

We have shown that the variable effect estimates obtained from experimental design sampling techniques are very useful in helping to rapidly find good solutions to large instances of the fixed charge capacitated multicommodity network design problem.

Guided Design Search generally produced better solutions, faster than the industry standard solver. These estimates help identify the significant variables and reveal problem characteristics. In the case of FCMD, the average effect of adding or deleting an arc is different information than that provided by a specific optimal solution and should be useful to network designers and managers who need some flexibility in their solutions.

The beneficial impacts to solution quality and time observed by setting only a small number of these significant variables lend support to the applicability of the sparsity of variable effects hypothesis to binary programs. Even though it was necessary to ignore the higher order (greater than 2) interaction effects in order to calculate main effects, GDS still yielded good results. The quantification and integration into GDS of two-level interaction effects is an area for future research. Another research path is more detailed analysis of the results in order to find better, problem-adaptive rules.

Acknowledgements

This research was supported by the School of Business and the Office of Research and Sponsored Programs at the University of Mississippi.

References

Ahuja, R.K., T.L. Magnanti, J.B. Orlin, 1993. *Network Flows – Theory, Algorithms, and Appications*. Prentice-Hall, Englewood Cliffs, NJ.

Balakrishnan, A., T.L. Magnanti, R.T. Wong, 1989. A Dual-Ascent Procedure For Large-Scale Uncapacitated Network Design. *Operations Research* **37** 5 716-740.

Barnhart, C., C.A. Hane, P.H. Vance, 2000. Using Branch-And-Price-And-Cut To Solve Origin-Destination Integer Multicommodity Flow Problems. *Operations Research* **48** 2 318-326.

Barr, R.S., B.L. Golden, J.P. Kelly, M.G.C. Resende, and W.R. Stewart, 1995. Designing and Reporting On Computational Experiments With Heuristic Methods. *Journal of Heuristics* **1** 1 9-32.

Berger, P.D. and R.E. Maurer, 2002. *Experimental Design with applications in management, engineering, and the sciences.* Duxbury – Thompson Learning.

Box, G.E.P. and R.D. Meyer, 1986. An Analysis For Unreplicated Fractional Factorials, Technometrics **28** 11-18.

Chinneck, J. W. and E. W. Dravnieks, 1991. Locating Minimal Infeasible Constraint Sets In Linear Programs. *ORSA Journal on Computing* **3** 2 157-168.

Craniac, T., M. Gendreau, J.M. Farvolden, 2000. A Simplex-Based Tabu Search Method For Capacitated Network Design. *INFORMS Journal on Computing* **12** 3 223-236.

Dahl, G. and M. Stoer, 1998. A Cutting Plane Algorithm For Multicommodity Survivable Network Design Problems. *INFORMS Journal on Computing* **10** 1 1-11.

DeVor, R., T. Chang, and J. Sutherland, *Statistical Quality Design and Control: Contemporary Concepts and Methods.* New York, NY: MacMillan Publishing Company, 1992.

Doucette, J., W. Grover, and R. Martens, 1999. Modularity And Economy-Of-Scale Effects In The Optimal Design Of Mesh-Restorable Networks, *Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering* 226-231.

Doucette J., and W. Grover, 2001. Comparison Of Mesh Protection And Restoration Schemes And The Dependency On Graph Connectivity, presented at the 3rd International Workshop on the Design of Reliable Communication Networks (DRCN), Budapest, Hungary.

Doucette, J., W. Grover and T. Bach, 2001. Bi-criteria Studies Of Mesh Network Restoration Path-Length Versus Capacity Tradeoffs, presented at OSA Optical Fiber Communications Conference, Anaheim, California.

Dunn, D., W. Grover, and M. Macgregor, 1994. Comparison Of K-Shortest Paths And Maximum Flow Routing For Network Facility Restoration. *IEEE Journal on Selected Areas in Communications* **12** 88-99.

Herrmann, J.W., G. Ioannou, I. Minis, J.M. Proth, 1994. A Dual Ascent Approach To The Fixed Charge Capacitated Network Design Problem. *Institute for Systems Research T.R.* **94** 29 1-30.

Holder, A., 2000. Designing Radiotherapy Plans with Elastic Constraints and Interior Point Methods. *Health Care and Management Science* **6** 1 5 - 16.

Holmberg, K and J. Hellstrand, 1998. Solving the Uncapacitated Network Design Problem By A Lagrangian Heuristic And Branch-And-Bound. *Operations Research* **46** 247-259.

Holmberg, K and D. Yuan, 2000. A Lagrangian Heuristic Based Branch-And-Bound For The Capacitated Network Design Problem. *Operations Research* **48** 3 461-481.

ILOG Optimization Suite. 2002. ILOG S. A., 9 Rue de Verdun, 94253 Gentilly Cedex, France, and ILOG, Inc., 1080 Linda Vista Ave., Mountain View, California 94043, USA.

Iraschko, R., M. MacGregor, and W. Grover, 1998. Optimal Capacity Placement For Path Restoration In STM Or ATM Mesh Survivable Networks. *IEEE/ACM Transactions on Networking* **6** 325-336.

Iraschko, R. and W. Grover, 2000. A highly Efficient Path-Restoration Protocol For Management Of Optical Network Transport Integrity. *IEEE Journal on Selected Areas in Communications* **18** 779-794.

Karger, D.R., 1999. Random Sampling In Cut, Flow, And Network Design Problems. *Mathematics of Operations Research* **24** 2 383-413.

Kennington, J., 1978. A Survey of Multicommodity Network Flows. *Operations Research* **26** 209-236.

Kennington, J. and K. Lewis. Generalized Networks: The Theory of Preprocessing and an Empirical Analysis. to appear in *INFORMS Journal on Computing*.

Kennington, J. and M. Lewis, 2001. The Path Restoration Version Of The Spare Capacity Allocation Problem With Modularity Restrictions: Models, Algorithms, And An Empirical Analysis. *INFORMS Journal on Computing* **13** 181-190.

Lamar, B.W., Y. Sheffi, W.B. Powell, 1990. A Capacity Improvement Lower Bound For Fixed Charge Network Design Problems. *Operations Research* **38** 4 704-710.

Linderoth, J.T. and M.W.P. Savelsbergh, 1999. A computational study of search strategies for mixed integer programming. *INFORMS Journal on Computing* **11** 2 173-187.

Luss, H., 1982. Operations Research And Capacity Expansion Problems: A Survey. *Operations Research* **30** 5 907-947.

Magnanti, T.L., R.T. Wong, 1986. Network Design And Transportation Planning: Models And Algorithms. *Transportation Science* **18** 1-55.

Mason, R., R. Gunst, and J. Hess, 1989. *Statistical Design and Analysis of Experiments*. New York, NY: John Wiley and Sons.

Montgomery, D.C. 1984. *Design and Analysis of Experiments*. New York, NY: John Wiley and Sons.

Ramaswami, R., 2002. Optical Fiber Communication: From Transmission To Networking. *IEEE Communications Magazine* 138-147.

Rardin, R. and U. Reha, 2001. Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial. *Journal of Heuristics* **7** 261-304.

Taguchi, G., A. Elsayed, and T. Hsiang, 1989. *Quality Engineering in Production Systems*. New York: McGraw-Hill.