This article was downloaded by: [128.32.10.230] On: 14 May 2018, At: 01:26
Publisher: Institute for Operations Research and the Management Sciences (INFORMS)
INFORMS is located in Maryland, USA

Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

50th Anniversary Article: Selection, Provisioning, Shared Fixed Costs, Maximum Closure, and Implications on Algorithmic Methods Today

Dorit S. Hochbaum,

To cite this article:
Dorit S. Hochbaum, (2004) 50th Anniversary Article: Selection, Provisioning, Shared Fixed Costs, Maximum Closure, and Implications on Algorithmic Methods Today. Management Science 50(6):709-723. https://doi.org/10.1287/mnsc.1040.0242

Full terms and conditions of use: http://pubsonline.informs.org/page/terms-and-conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

© 2004 INFORMS

Please scroll down for article—it is on subsequent pages

INFORMS is the largest professional society in the world for professionals in the fields of operations research, management science, and analytics.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

RIGHTSLINK

MANAGEMENT SCIENCE
Vol. 50, No. 6, June 2004, pp. 709–723
ISSN 0025-1909 | EISSN 1526-5501 | 04 | 5006 | 0709

informs

DOI 10.1287/mnsc.1040.0242
© 2004 INFORMS

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

50th Anniversary Article

Selection, Provisioning, Shared Fixed Costs, Maximum Closure, and Implications on Algorithmic Methods Today

Dorit S. Hochbaum
Department of Industrial Engineering and Operations Research and Walter A. Haas School of Business, University of California, Berkeley, California 94720, hochbaum@ieor.berkeley.edu

Motivated by applications in freight handling and open-pit mining, Rhys, Balinski, and Picard studied the problems of selection and closure in papers published in *Management Science* in 1970 and 1976. They identified efficient algorithms based on linear programming and maximum-flow/minimum-cut procedures to solve these problems. This research has had major impact well beyond the initial applications, reaching across three decades and inspiring work on numerous applications and extensions. The extensions are nontrivial optimization problems that are of theoretical interest. The applications ranged from evolving technologies, image segmentation, revealed preferences, pricing, adjusting utilities for consistencies, just-in-time production, solving certain integer programs in polynomial time, and providing efficient 2-approximation algorithms for a wide variety of hard problems. A recent generalization to a convex objective function has even produced novel solutions to prediction and Bayesian estimation problems. This paper surveys the streams of research stimulated by these papers as an example of the impact of *Management Science* on the optimization field and an illustration of the far-reaching implications of good original research.

Key words: parametric cut; minimum-cost flow; financial risk; medical prognosis

1. Introduction

In 1970 Rhys and Balinski independently published papers in *Management Science* motivated by the *freight-handling terminals* problem, which concerns setting up links between terminals for the purpose of establishing transportation links, such as airline travel, between cities. The opening of each link is associated with profit or benefit generated from the operation of that link. However, to operate a link it is necessary to construct terminals at both ends, and thus incur the fixed cost of construction on both ends.

This problem would have been resolved easily if each link were independent from the others. However, once a terminal exists it can be used for links with other cities that already have terminals. So it may well happen that while the cost of a pair of terminals exceeds the potential benefit of the link between the pair, these terminals will be constructed nevertheless because the fixed costs of constructing the terminals are shared among several links. Because of this Rhys labelled it the *shared fixed-cost* problem. We follow Balinski and refer to it as the *selection problem*.

The selection problem can be thought of formally as follows: Each link is a *set* consisting of two cities associated with a benefit value. Each terminal is an *item* associated with a cost value. The objective is to find a collection of sets so that their total benefit minus the cost of the elements in their *union* is maximized. Although in the specific scenario of freight terminals each set consists of two elements, in general, sets could be of arbitrary size.

Lawler (1976), in his classical book, described the selection problem—which he termed the *provisioning* problem—by a scenario resembling the knapsack problem, which is well known to be NP-complete. This scenario has you going on a hiking trip with a knapsack to carry. Because your capability of carrying heavy weight is limited, you want to decide which items to pack based on their usefulness and utility. Many individual items are useless unless taken in combination with other items. For example, taking a bottle of wine without an opener is not of much use. Another example is soup for your meal. For that purpose you will require a few different items: canned soup, an opener, a spoon, and a bowl, and possibly a portable stove, pot, and fuel. Obviously the stove, pot, and fuel could have other uses as well. Because of the similarity of description, most people hearing the provisioning problem description for the first time would guess it is an intractable problem like the knapsack

RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

problem, yet, the selection problem is surprisingly polynomial time solvable.

Given a set of items $\{1, \ldots, n\}$ each having an associated cost, c_j, and sets of items $S_i \subseteq \{1, \ldots, m\}$ each having benefit b_i, the problem is to maximize the net benefit, which is the total benefit of the sets selected minus the total cost of items selected. A "selection" corresponds to a collection of sets $\{S_i \mid i \in J\}$, and the selected items are $\{\bigcup S_i, i \in J\}$.

To formulate the selection problem we define two types of binary variables,

$$v_j = \begin{cases} 1 & \text{if item } j \text{ is included} \\ 0 & \text{otherwise,} \end{cases}$$

$$u_i = \begin{cases} 1 & \text{if set } i \text{ is selected} \\ 0 & \text{otherwise,} \end{cases}$$

which enables us to specify the following optimization problem:

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} b_i u_i - \sum_{j=1}^{n} c_j v_j \\ \text{subject to} \quad & u_i \le v_j \quad \forall j \in S_i, \quad i = 1, \ldots, m \\ & u_i, v_j \in \{0,1\}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n. \end{aligned}$$

The constraints enforce the restriction that the benefit of a set cannot be enjoyed unless all its items have been paid for. The work of Rhys (1970) and Balinski (1970) was to show that the selection problem can be solved as a minimum-cut problem on a certain bipartite network.

Picard (1976) generalized the selection problem to the closure problem. The major application motivating Picard's work was the *open-pit mining* problem. The mining industry at the time, and to some extent still today, was developing solution methods independently of the operations research community. Picard's contribution was to unearth the link between the two communities, and make possible the use of efficient algorithms of the maximum-flow–minimum-cut problem in mining. In the opposite direction, Picard's discovery made possible the use of algorithms developed for the mining and closure problem for the development of new maximum-flow algorithms. Indeed, the pseudoflow algorithm for maximum flow (Hochbaum 2002) is patterned after the algorithm of Lerchs and Grossman (1965) used by the mining industry since the 1960s.

Open-pit mining is a surface mining operation in which blocks of earth are extracted from the surface to retrieve the ore contained within. During the mining process, the surface of the land is being continuously excavated, and a deeper and deeper pit is formed until the operation terminates. The final contour of this pit mine is determined before mining operation begins. To design the optimal pit—one that maximizes profit—the entire area is divided into blocks, and the value of the ore in each block is estimated by using geological information obtained from drill cores. Each block has a weight associated with it that represents the value of its ore minus the cost involved in removing the block. While trying to maximize the total weight of the blocks to be extracted, there are also contour constraints that have to be observed. These constraints specify the slope requirements of the pit and precedence constraints that prevent blocks from being mined before others on top of them. Subject to these constraints, the objective is to mine the most profitable set of blocks.

The open-pit mining problem can be represented on a directed graph $G = (V, A)$. Each block i corresponds to a node with a weight b_i representing the net value of the individual block. The net value is computed as the assessed value of the ore in that block, from which the cost of extracting that block alone is deducted. The weight can therefore be positive or negative or zero. There is a directed arc $(i, j) \in A$ from node i to node j if block i cannot be extracted before block j, which is in a layer above block i. This precedence relationship is determined by the engineering slope requirements. Suppose block i cannot be extracted before block j, and block j cannot be extracted before block k. By transitivity this implies that block i cannot be extracted before block k. It is therefore necessary only to include immediate successors as arcs in the graph. The decision of which blocks to extract to maximize profit is equivalent to finding a maximum weight set of nodes in the graph such that all successors of all nodes are included in the set.

A set of nodes $D \subseteq V$ in a directed graph $G = (V, A)$ is called *closed* if all successors of nodes in D are also in D. In other words, there is no arc from a node in D to a node outside of D. The open-pit mining problem can be modeled as the maximum-*closure* problem stated formally: Given a directed graph $G = (V, A)$ and node weights (positive or negative) b_i for all $i \in V$, find a closed subset of nodes $V' \subseteq V$ such that $\sum_{i \in V'} b_i$ is maximum.

To formulate the maximum-closure problem, let x_j be a binary variable that is 1 if node j is in the closure, and 0 otherwise. Define b_j to be the weight of the node or the net benefit derived from the corresponding block. Then we can write the following optimization problem:

$$\begin{aligned} \max \quad & \sum_{j \in V} b_j \cdot x_j \\ \text{subject to} \quad & x_j - x_i \ge 0 \quad \forall (i, j) \in A \\ & 0 \le x_j \le 1 \quad \text{integer} \quad j \in V. \end{aligned}$$

The maximum-closure problem is essentially a nonbipartite version of the selection problem in the sense

RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

that there is no distinction between items and sets. Picard showed that this generalization is also solvable as a minimum-cut problem on some related graph.

The papers by Balinski (1970), Rhys (1970), and Picard (1976) on the closure problem have had broad, often unintended, impacts on theoretical and applied research developments, which still continue today. In this paper we present a list of problems that extend the selection and closure problems in several ways and a host of applications, some of which have been discovered recently, that can be solved effectively thanks to the techniques initially outlined in the three pioneering papers of Balinski, Rhys, and Picard. These extension problems have all been shown to be solvable using a minimum-cut algorithm as a main subroutine. Indeed, this nontrivial equivalence to the minimum-cut problem is one of the surprising and useful features of the works of Rhys, Balinski, and Picard. This equivalence extends to generalization of the closure. Minimum-cut models thus yield particularly powerful algorithms for a vast range of applications.

Since the selection/closure problems and their extensions have a broad range of applications in numerous areas, the ability to solve these problems efficiently using so-called combinatorial algorithms of minimum cut is of substantial computational significance. For many of these applications getting quick solutions is essential in practical contexts. Combinatorial algorithms—which do not apply algebraic operations that may lead to round-off errors, but rather require only simple additions and multiplications—are particularly desirable for their simplicity and efficiency in such circumstances.

Table 1 summarizes the various problem definitions, their formulations, and representative applications. All variables in the formulations, denoted by x_i or z_{ij}, are integers. So, $0 \le x_i \le 1$ is equivalent to stating that x_i is binary. The formulations of all problems other than IPM and IP2 are defined on a graph

Table 1 Extensions of the Selection/Closure Problems and Their Formulations

Problem	Formulation	Reference	Application
Selection	$\min \sum_{i\in V} w_i x_i$ s.t. $x_i - x_j \le 0$ for $i \in V_1, j \in V_2, (i,j) \in A$ $0 \le x_i \le 1$ for $i \in V$	Rhys (1970), Balinski (1970)	Freight terminals
Closure	$\max \sum_{i\in V} w_i x_i$ s.t. $x_i - x_j \le 0$ for $(i,j) \in A$ $0 \le x_i \le 1$ for $i \in V$	Picard (1976)	Open-pit mining
Convex closure	$\min \sum_{i\in V} w_i(x_i)$ s.t. $x_i - x_j \le 0$ for $(i,j) \in A$ $l_i \le x_i \le u_i$ for $i \in V$	Hochbaum and Queyranne (2003)	Prediction and Bayesian estimation
s-excess	$\max \sum_{i\in V} w_i x_i - \sum_{(i,j)\in A} u_{ij} z_{ij}$ s.t. $x_i - x_j \le z_{ij}$ for $(i,j) \in A$ $0 \le x_i \le 1$ for $i \in V$ $0 \le z_{ij} \le 1$ for $(i,j) \in A$	Hochbaum	Max-flow Cell selection
Convex *s*-excess	$\min \sum_{i\in V} w_i(x_i) - \sum_{(i,j)\in A} u_{ij} z_{ij}$ s.t. $x_i - x_j \le z_{ij}$ for $(i,j) \in A$ $l_i \le x_i \le u_i$ for $i \in V$ $0 \le z_{ij} \le U_{ij}$ for $(i,j) \in A$	Hochbaum	Image segmentation
Monotone integer programming (IPM)	$\min \sum_{i=1}^{n} w_i x_i$ s.t. $a_k x_{i_k} + b_k x_{j_k} \le c_k$ for $k = 1, \ldots, m$ $l_i \le x_i \le u_i$ for $i = 1, \ldots, n$ a_k and b_k of opposite signs for $k = 1, \ldots, m$	Hochbaum and Naor (1994)	Bipartite vertex cover
(Nonmonotone) IP2	$\min \sum_{i=1}^{n} w_i x_i$ s.t. $a_k x_{i_k} + b_k x_{j_k} \le c_k$ for $k = 1, \ldots, m$ $l_i \le x_i \le u_i$ for $i = 1, \ldots, n$	Hochbaum et al. (1993)	Max-clique Vertex cover 2-SAT
2var	$\min \sum_{i=1}^{n} w_i x_i + \sum U_k z_k$ s.t. $a_k x_{i_k} + b_k x_{j_k} \le c_k + z_k$ for $k = 1, \ldots, m$ $l_i \le x_i \le u_i$ for $i = 1, \ldots, n$ $0 \le z_k \le \gamma_k$ for $k = 1, \ldots, m$	Hochbaum	Forestry Postal services location
Convex DMCNF	$\min \sum_{i=1}^{n} w_i(x_i) + \sum U_{ij}(z_{ij})$ s.t. $x_i - x_j \le c_{ij} + z_{ij}$ for $(i,j) \in A$ $l_i \le x_i \le u_i$ for $i \in V$ $0 \le z_{ij}$ for $(i,j) \in A$	Ahuja et al. (2004)	Project Management Dial-a-ride

RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

$G = (V, A)$ (which is bipartite for the case of selection only). The functions $w_i()$ and $U_{ij}()$ $U_k()$ are assumed to be convex.

The remainder of the paper is organized as follows: Section 2 shows the theoretical equivalence of selection, closure, and various extension problems to the minimum-cut problem. Section 3 discusses a range of applications that can be modeled as generalizations of selection and closure problems. We conclude in §4 with some observations on the implications of this historical survey for future research.

2. The Equivalence to Minimum Cut

2.1. The Selection Problem

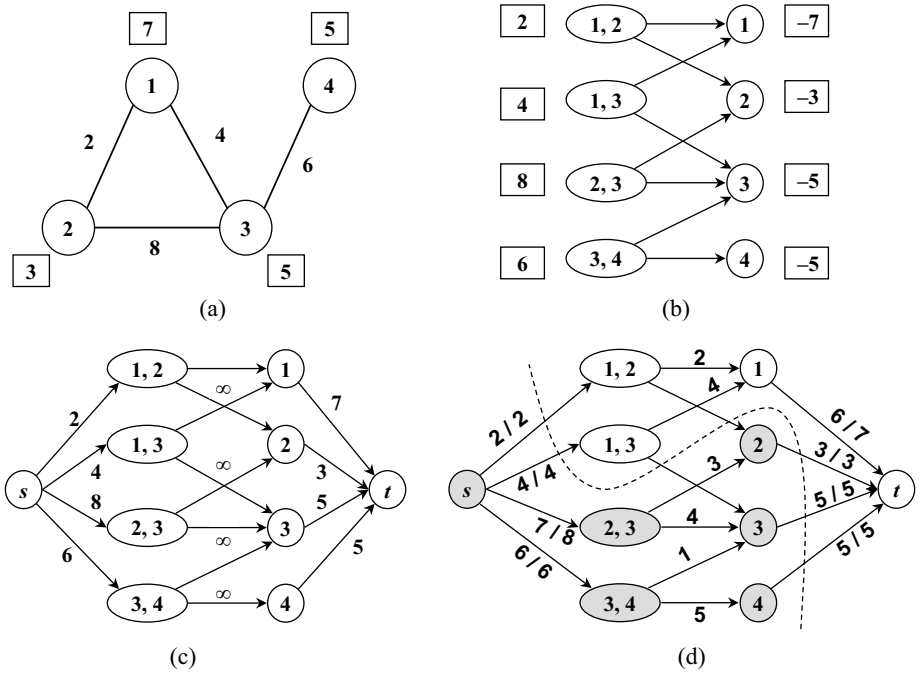
The linear programming formulation of the selection problem in Table 1 was given by Rhys (1970). This formulation has in each constraint at most one coefficient 1 and one coefficient -1, which guarantees that the constraint matrix is totally unimodular. Therefore each basic solution is integer and in particular there is an optimal linear programming solution that is integer.

Balinski showed how this formulation can be solved more efficiently than with linear programming by finding a minimum s,t-cut in a suitable graph. The graph constructed is a bipartite graph with nodes corresponding to sets V_1 on one side each with benefit weight b_i, and nodes corresponding to items V_2 on the other, each with cost weight $-c_j$. The set of arcs is $A = \{(i, j) \mid j \in S_i\}$, all of which are assigned infinite capacity. It is easy to see that the maximum-closure problem defined on this node-weighted bipartite graph is the selection problem. We now append the graph with a source s and a sink t and replace the weights by adding arcs from s to i with capacity b_i, and from j to t with capacity c_j. An s,t-cut separating source s from sink t is a partition of the set of nodes $\{s\} \cup V_1 \cup V_2 \cup \{t\}$ to two subsets S and $T = \overline{S}$, so that $s \in S$ and $t \in T$. The capacity of a s,t-cut (S, T) is $C(S, T) = \sum_{i \in S,\, j \in T} u_{ij}$, where u_{ij} are the arc capacities. A s,t-cut of minimum capacity is a minimum s,t-cut, or simply a minimum cut.

The equivalence of the selection problem to a minimum s,t-cut problem is demonstrated for a freight terminals example similar to the one given by Rhys (1970) in Figure 1(a). There are four potential terminal locations and four links with the costs and benefits indicated. Figure 1(b) shows the corresponding bipartite network. Figure 1(c) shows the bipartite network in which the maximum-flow and minimum-cut problems are solved. The set of shaded nodes in Figure 1(d) is the source set of the minimum cut, which contains the optimal selection of terminals—Terminals 2, 3 and 4. The value of the optimal flow on each arc of finite capacity is indicated by flow/capacity.

It is now shown that in general the source set of a minimum cut is the union of the sets and elements of an optimal selection. Since arcs (i, j) have infinite capacity, a feasible selection corresponds to a *finite-capacity* cut. This is because the inclusion of a set in the source set implies that all of its elements are in the source set as well. For a finite cut (S, T) the corresponding selection $S\backslash\{s\}$ is the collection of sets $S \cap V_1$ and the union of elements in these sets

Figure 1 **The Terminal Selection Problem**



RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

$S \cap V_2$. The net benefit of this selection is $\text{NetBen}(S) = \sum_{i \in S \cap V_1} b_i - \sum_{j \in S \cap V_2} c_j$.

$$\begin{aligned} C(S,T) &= \sum_{i \in T \cap V_1} b_i + \sum_{j \in S \cap V_2} c_j \\ &= \sum_{i \in V_1} b_i - \sum_{i \in S \cap V_1} b_i + \sum_{j \in S \cap V_2} c_j \\ &= \sum_{i \in V_1} b_i - \text{NetBen}(S). \end{aligned}$$

Since $\sum_{i \in S \cap V_1} b_i$ is a constant, a cut of minimum capacity corresponds to a selection maximizing net benefit.

2.2. The Maximum-Closure Problem

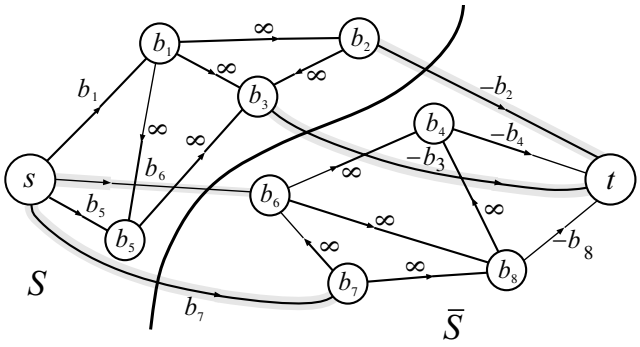
Picard (1976) formulated the closure problem as a quadratic 0-1 problem and showed how the latter can be solved as minimum-cut problem. Although this could have also been concluded from the works of Rhys (1970) and Balinski (1970), as we see next, the relationship was not fully understood at the time.

The selection problem is clearly a special case of the closure problem. By casting the problem on the direct bipartite graph constructed as in Figure 1(b), any selection is a closed set. Recall that in a directed graph $G = (V, A)$, a subset of the nodes $D \subset V$ is called *closed* if all successors of D are contained in D, or if the set of nodes reachable from D is D.

On the other hand, and this justifies the opening statement, any closure problem defined on a graph $G = (V, A)$ is also a special case of a selection problem defined on a bipartite graph. To see this, we let all nodes of positive weight in the closure problem be the set of nodes V_1, and all nodes of negative weight be the set V_2. There is an arc from $i \in V_1$ to $j \in V_2$ if j is in the closure of i, or, in other words, if there is a path in G from i to j. We refer to this construction as *bipartizing* the graph. Johnson (1968) seems to be the first researcher who demonstrated that the maximum-closure problem is equivalent to the selection problem (maximum closure on bipartite graphs), and that the selection problem is solvable by a transportation algorithm. So, interestingly, Johnson's algorithm—devised for the maximum-closure problem and motivated specifically by the open-pit mining application—was based on casting the problem first as a selection problem. In general, bipartizing is not computationally practical as it increases the number of arcs in the graph.

Picard (1976) demonstrated that a minimum-cut algorithm on a related graph solves the maximum-closure problem. The *related graph* is constructed by adding a source and a sink node, s and t, $\widetilde{V} = V \cup \{s, t\}$. Let $V^+ = \{j \in V \mid b_j > 0\}$, and $V^- = \{j \in V \mid b_j < 0\}$. The set of arcs in the related graph, $\widetilde{A}$, is the set A appended by arcs $\{(s, v) \mid v \in V^+\} \cup \{(v, t) \mid v \in V^-\}$.

Figure 2 A Maximum Closure and Minimum Cut in a Closure Graph



The capacity of all arcs in A is set to ∞, and the capacity of all arcs adjacent to source or sink is $|b_v|$: $u_{s,v} = b_v$ for $v \in V^+$ and, $u_{v,t} = -b_v$ for $v \in V^-$. The source set of a minimum cut separating s from t is also a maximum closure in the graph. The source set is obviously closed as the minimum cut must be finite and thus cannot include any arcs of A.

Let a finite cut be $(S, \bar{S})$ in the graph $\widetilde{G} = (\widetilde{V}, \widetilde{A})$. Let $B = \sum_{j \in V} b_j$ be the sum of all weights and, thus, a fixed constant. The capacity of the cut $(S, \bar{S})$ is,

$$\begin{aligned} \sum_{j \in V^- \cap S} |b_j| + \sum_{j \in V^+ \cap \bar{S}} b_j &= \sum_{j \in V^- \cap S} |b_j| + \sum_{j \in V} b_j - \sum_{j \in V^+ \cap S} b_j \\ &= B - \sum_{j \in S} b_j. \end{aligned}$$

Hence minimizing the cut capacity is equivalent to maximizing the total sum of weights of nodes in the source set of the cut, which is closed. A schematic description of a closure graph related to a maximum-closure problem, where the source set of the minimum cut is the maximum closed set, is depicted in Figure 2.

2.3. The Convex Cost Closure Problem

A common problem in statistical estimation is that observations do not satisfy preset ranking order requirements. The challenge is to find an adjustment of the observations that fits the ranking order constraints and minimizes the total deviation penalty. The deviation penalty is a convex function of the fitted values. This problem motivated the introduction of the convex cost closure (CCC) problem in (Hochbaum and Queyranne 2003).

The CCC problem is defined formally on a directed graph $G = (V, A)$ and convex functions $f_j()$ associated with each node $j \in V$. The formulation of the CCC problem is then,

$$\begin{aligned} \text{(CCC)} \qquad \min \quad & \sum_{j \in V} f_j(x_j) \\ \text{subject to} \quad & x_i - x_j \geq 0 \quad \forall (i,j) \in A \\ & l_j \leq x_j \leq u_j \quad \text{integer } j \in V. \end{aligned}$$

RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

This problem generalizes the closure problem in that the variables assume a range of integer values (rather than being binary) and the objective is convex.

The threshold theorem of Hochbaum and Queyranne (2003) is the key result that leads to an efficient algorithm by reducing the convex problem to its binary counterpart—the minimum-closure problem.

To sketch the main idea of the theorem we first note that one can extend all the functions $f_i()$ so that they are convex in the range $[l, u]$ for $l = \min_i l_i$, $u = \max_i u_i$. Let α be scalar and w_i be the derivative or subgradient of f_i at α, $w_i = f_i'(\alpha) = f_i(\alpha+1) - f_i(\alpha)$. Let $G_\alpha = (V, A)$ be a closure graph with node weights w_i. The threshold theorem states that for the optimal closure in G_α, S_α, the optimal values of the variables for the convex problem x_j^* are $>\alpha$ if $j \in S_\alpha$, or $\leq\alpha$ otherwise. An example of such graph is given for the more general convex s-excess or image-segmentation problem in Figure 4 of §3.6.

By repeated applications of the minimum-closure algorithm on the graph G_α for a range of values of α in $[l, u]$, we obtain a partition of the set of variables and of the interval $[l, u]$ into up to n subsets and subintervals where each subinterval contains the optimal value of one subset of variables. Moreover, it is shown in Hochbaum and Queyranne (2003) that this partition can be achieved with a parametric minimum-cut procedure where α is the parameter.

The procedure used to solve the parametric minimum-cut problem is a generalization of a procedure devised by Gallo et al. (1989) for linear functions of the parameter, which are based on the push-relabel algorithm of Goldberg and Tarjan (1988). The generalization for any monotone functions is described in Hochbaum (2003) and in Hochbaum (2002) for both the push-relabel algorithm and the pseudoflow algorithm. The algorithm requires at each iteration finding the integer minima of the convex functions, which is accomplished with binary search in $O(n \log U)$ steps. Note that the CCC problem generalizes the minimum-cut problem and it is at least as hard as minimization of n convex functions over bounded intervals. Hence the run time cannot be improved unless the respective run times of the minimum-cut problem and minimizing convex functions can be improved. A proof that optimization involving nonlinear nonquadratic functions cannot be accomplished in strongly polynomial time, and thus cannot be substantially improved beyond the bound given, is provided in Hochbaum (1994).

2.4. Integer Programming on Monotone Inequalities

An inequality in two variables $a_i x_{j_i} - b_i x_{k_i} \geq c_i$ is said to be *monotone* if both a_i and b_i are nonnegative or nonpositive. That is, the coefficients of the two variables are of opposite signs. The formulation of integer programming on monotone inequalities is,

$$\text{(IPM)} \quad \min \quad \sum_{j=1}^{n} w_j x_j$$

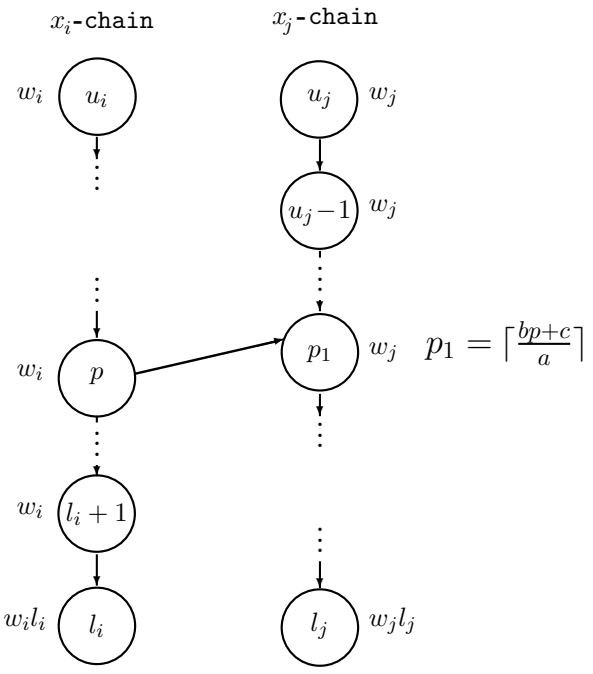
$$\text{subject to} \quad a_i x_{j_i} - b_i x_{k_i} \geq c_i \quad (i = 1, \ldots, m)$$

$$l_j \leq x_j \leq u_j, \quad x_j \text{ integer } (j = 1, \ldots, n),$$

where a_i, b_i, c_i $(i = 1, \ldots, m)$, and w_j $(j = 1, \ldots, n)$ are rational, and coefficients a_i and b_i $(i = 1, \ldots, m)$ are of the same sign.

Hochbaum and Naor (1994) devised an algorithm to solve IPM by casting the problem as a closure problem on a graph, schematically depicted in Figure 3. A directed closure graph $G = (V, A)$ is created where, for each variable value x_j and value p in the interval $[l_j, u_j]$, there is a node representing it. In the sequence of nodes representing the values of variable x_j there is an arc $(p, p-1)$ from each node representing the value p in the range to the node representing the value $p-1$. The node representing l_j has an arc of infinite capacity directed to it from the source node s. Thus all l_j nodes are in the closed set containing the source s. The monotone inequalities are represented by arcs. For each potential value p of variable x_{k_i}, all inequalities in which x_{k_i} appears with negative coefficient impose a minimum value on the variable x_{j_i}

Figure 3 **Representing the Inequality $ax_i - bx_j \geq c$ Between the Chains for x_i and x_j**



RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

that appears in the same inequality with a positive coefficient,

$$x_{j_i} \geq \left\lceil \frac{b_i p + c_i}{a_i} \right\rceil = p_1.$$

This inequality is represented by an arc going from node p of x_{k_i} to node p_1 of x_{j_i}. If $p_1 > u_{j_i}$, then the value p of the variable x_{k_i} is infeasible, and the upper bound of x_{k_i} is reset to $p-1$. A closed set containing s corresponds to a feasible solution to IPM where the variable x_j assumes the value of the largest node representing it in the closed set.

The nodes are now assigned weights as follows: Node l_j of variable x_j is assigned weight $w_j l_j$, and all other nodes representing variable x_j are assigned the weight w_j. A minimum-weight closed set corresponds to an optimal solution to the minimization problem IPM by setting the variables as the largest value in the chain that is in the closed set. Thus the integer programming on monotone inequalities is solved in the complexity of minimum cut on a graph with $O(nU)$ nodes and $O(mU)$ arcs $-O(mnU^2 \log(Un^2/m))$ for U the largest variable range, $U = \max_{j=1}^{n}\{\lfloor u_j \rfloor - \lceil l_j \rceil\}$. It should be noted that the dependence of the complexity on U is *pseudopolynomial*. Nevertheless it is not possible to replace this term by $\log U$ or eliminate it altogether, as the problem of finding a feasible integer solution on a set on monotone inequality is NP-hard. For more details on this issue the reader is referred to Hochbaum and Naor (1994).

2.4.1. Nonlinear Monotone Integer Programming. The objective function in this problem is nonlinear separable, with arbitrary functions $w_j()$.

$$\begin{aligned} \text{(nonlin-IPM)} \qquad \min \quad & \sum_{j=1}^{n} w_j(x_j) \\ \text{subject to} \quad & a_i x_{j_i} - b_i x_{k_i} \geq c_i \quad (i = 1, \ldots, m) \\ & l_j \leq x_j \leq u_j, \\ & x_j \text{ integer } (j = 1, \ldots, n). \end{aligned}$$

The same algorithm used for IPM applies here with the modification of the node weight assignments only. Node l_j of variable x_j is assigned weight $w_j(l_j)$. For the node representing the value p of variable x_i, the weight of the node is $w_j(p) - w_j(p-1)$. The various nodes in the chain of one variable x_i may have different signs, unlike the linear case of the problem IPM. The complexity for solving the problem is the same as that for the linear IPM, $O(mnU^2 \log(n^2 U/m))$.

2.5. The Maximum s-Excess Problem

The s-excess problem is a variant of the maximum-closure problem with a *relaxation* of the closure requirement: Nodes that are successors of other nodes in S (i.e., that have arcs originating from node of S to these nodes) may be excluded from the set but at a penalty that is equal to the capacity of those arcs. In a closure graph these arcs are of infinite capacity. For the s-excess problem the arcs have finite capacities representing the penalties for violating the closure requirement.

The maximum s-excess problem is defined on a directed graph $G = (V, A)$, with node weights (positive or negative) w_i for all $i \in V$, and nonnegative arc weights u_{ij} for all $(i, j) \in A$. The objective is to find a subset of nodes $S \subseteq V$ such that $\sum_{i \in S} w_i - \sum_{i \in S,\, j \in \bar{S}} u_{ij}$ is maximum.

A generalized form of Picard's (1976) theorem (see §2.2) showing that the closure problem is equivalent to the minimum-cut problem was proved for the s-excess problem in Hochbaum (2002). The idea here is to construct a graph as for the closure problem except that the arc capacities not adjacent to source and sink for (i, j) in A are the respective weights u_{ij}. The source set of a minimum cut in the graph created is shown to be the maximum s-excess set. The interested reader is referred to Hochbaum (2002) for details.

The s-excess problem has appeared in several forms in the literature: The boolean quadratic minimization problem with all of the quadratic terms having positive coefficients is a restatement of the s-excess problem. More closely related is the feasibility condition of Gale (1957) for a network with supplies and demands, or Hoffman's (1960) for a network with lower and upper bounds. Verifying feasibility is equivalent to ensuring that the maximum s-excess is zero in a graph with node weights equal to the respective supplies and demands with opposite signs; if the s-excess is positive, then there is no feasible flow satisfying the supply and demand balance requirements. This problem also appeared under the names *maximum-blocking cut* or *maximum-surplus cut* in Radzik (1993).

2.6. The Convex s-Excess Problem

The convex s-excess problem is a generalization of the s-excess problem in allowing the variables to take nonbinary integer values and variable weights $f_j()$ that are convex functions.

$$\begin{aligned} \text{(Convex } s\text{-excess)} \quad \min \quad & \sum_{j \in V} f_j(x_j) + \sum u_{ij} z_{ij} \\ \text{subject to} \quad & x_i - x_j \leq z_{ij} \quad \text{for } (i, j) \in A \\ & \bar{u}_j \geq x_j \geq l_j \quad j = 1, \ldots, n \\ & z_{ij} \geq 0 \quad (i, j) \in A. \end{aligned}$$

A threshold theorem for the convex s-excess problem generalizing the one in Hochbaum and Queyranne (2003) was proved in Hochbaum (2001b).

RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

The essence of the theorem is to reduce the convex s-excess problem to the s-excess problem on binary variables. In Hochbaum (2001b) it was shown that generalizing the procedure used for the convex-closure problem, one can solve the problem by using the threshold theorem, solving a parametric minimum-cut procedure. The total complexity of solving the convex s-excess problem is therefore $O(mn\log(n^2/m) + n\log U)$, the same as the convex-closure problem. Similar arguments to those given in §2.3 demonstrate that this complexity expression is the best that can be achieved for the problem.

2.7. Monotone Integer Programs with Three Variables per Inequality

Monotone integer programs on three variables per inequality (2var) are characterized by constraints of the form $ax - by \le c + z$, where a and b are nonnegative and the variable z appears only in that constraint. The direction of the inequality is immaterial and the coefficients a and b can assume any real value as long as $b \ge 1$. (Otherwise it would always be possible to calibrate the coefficients so that the coefficient of z is equal to 1.) Since any integer programming problem can be expressed in three variables per inequality, the restriction that z appears in one constraint limits the applicability to a strict subset of integer programs. The objective function in these integer programming problems is unrestricted except that the functions of z must be convex. This class of problems is of particular interest because of the large variety of problems that are formulated as 2var.

A 2var problem is solved as an s-excess problem on a graph with $O(nU)$ nodes and $O(mnU)$ arcs. For details on this class of problems and a review of applications the reader is referred to Hochbaum (2002). The same type of graph setup works to solve the problem with the same complexity even for nonlinear objective function, $\min \sum_{i=1}^{n} w_i(x_i) + \sum e_k(z_k)$, where $w_i()$ are general nonlinear functions and $e_k()$ are convex functions.

2.8. Convex Dual of Minimum-Cost Network Flow

The formulation of Convex DMCNF has each structural constraint involving a pair of variables and possibly a third variable that appears in that constraint only, and allows for the objective function to be convex in all variables. The latter feature generalizes the convex s-excess problem. Let the set of constraints with three variables be E_1 and the set of constraints with two variables be E_2. Let the set of variables be $V = \{1, \ldots, n\}$. The set of constraints E is partitioned into two subsets $E = E_1 \cup E_2$, with $|E_1| = m_1$, $|E_2| = m_2$, $|V| = n$, and $|E| = m_1 + m_2 = m$. Let the functions $uu_{ij}()$ be convex. The problem addressed is,

$$
\begin{aligned}
\text{(DMCNF)} \quad \min \quad & \sum_{j=1}^{n} w_j(x_j) + \sum_{(i,j)\in E_1} uu_{ij}(z_{ij}) \\
\text{subject to} \quad & x_i - x_j \le c_{ij} + z_{ij} \quad \text{for } (i,j) \in E_1 \\
& x_i - x_j \le c_{ij} \quad \text{for } (i,j) \in E_2 \\
& l_j \le x_j \le u_j \quad j = 1, \ldots, n \\
& 0 \le z_{ij} \le \gamma_{ij} \quad \text{for } (i,j) \in E_1 \\
& x_j \text{ integer for all } j = 1, \ldots, n \\
& z_{ij} \text{ integer for all } (i,j) \in E_1.
\end{aligned}
$$

The dual of DMCNF with linear objective function $\sum_{j=1}^{n} w_j x_j + \sum_{(i,j)\in E_1} uu_{ij} z_{ij}$ is the minimum-cost network flow problem (Flow).

$$
\begin{aligned}
\text{(Flow)} \quad \min \quad & \sum_{ij\in E_1\cup E_2} c_{ij} y_{ij} + \sum_{i=1}^{n} u_i \alpha_i + \sum_{ij\in E_1} \gamma_{ij} \delta_{ij} \\
\text{subject to} \quad & -\sum_{k} y_{ik} + \sum_{k} y_{ki} - \alpha_i \le w_i \quad i \in V \\
& \bar{u}_{ij} \ge y_{ij} - \delta_{ij} \ge 0 \quad (i,j) \in E_1 \\
& y_{ij} \ge 0 \quad (i,j) \in E_1 \cup E_2 \\
& \delta_{ij} \ge 0 \quad (i,j) \in E_1 \\
& \alpha_i \ge 0 \quad i = 1, \ldots, n.
\end{aligned}
$$

Flow is the formulation of a network flow problem on a network with n nodes—one per structural constraint and a dummy node, r, serving as a root. The variable α_i represents the flow from node i to the root. The inflow to node i exceeds the outflow by at most w_i. This quantity is assigned as capacity to arcs going from node i to the root. The costs of these arcs are u_i. The costs of all other arcs not adjacent to root is c_{ij}. For each such arc that belongs to E_1 there is an additional parallel arc of unbounded capacity with a cost of $c_{ij} + \gamma_{ij}$. The amount of flow on this parallel arc is δ_{ij}; this flow is positive only if the flow on the first arc has reached its capacity $\bar{u}_{ij}$.

Problem DMNCF is a monotone 2var problem that has a totally unimodular constraint matrix—the coefficients of x_i and x_j in the constraints are 1 and -1. This property allows us to solve the problem in polynomial time that depends on $\log U$ rather than on U, for $U = \max_{j=1,\ldots,n}\{|u_j - l_j|\}$. This is done with the proximity-scaling algorithm of Hochbaum and Shanthikumar (1990), which is particularly efficient for problems with small subdeterminants values. That algorithm reduces the problem to a logarithmic number of scaled problems where for each one the range U is at most $O(n^2)$ units, each of which is a 2var problem solvable by minimum-cut algorithm. This construction is described in Ahuja et al. (2004).

A different and more efficient algorithm for the same problem that does not rely on reduction to minimum cuts was devised by Ahuja et al. (2003).

RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

3. Applications

Much of the interest in selection and closure problems stems from the wide range of practical applications that can be modeled with them. In this section we discuss some of the most important ones.

3.1. Optimal Kit of Parts and Tools as Selection Problem

Mamer and Smith (1982) used the selection problem to solve a problem of determining the optimal kit of parts and tools for on-site equipment repairs. The model they presented generalized previous approaches in that part demands were not assumed to be independent and allowed for the requirement of various numbers of parts, as well as multiple part types per job. Handling this dependence was made possible by using the features of the selection problem. Even though the model used by Mamer and Smith was more realistic than previous models of this problem, they were also able to solve it by a more efficient algorithm than used previously—the minimum-cut algorithm.

3.2. Prediction, Bayesian Estimation, and Other Applications of Convex Closure

3.2.1. Multistage Production/Inventory. Maxwell and Muckstadt (1983) considered *nested power-of-two policies* in a *multistage production/inventory* problem. In this continuous-time deterministic model demand for end products arises at a constant rate. Intermediate products are consumed in the production of other products, as reflected by a directed graph (V, A). For given positive inventory-related holding costs g_j and production setup costs K_j, the problem is to find production intervals $T_j = T_0 2^{k_j}$, with k_j integer, that are *nested*. That is $T_i \le T_j$ for $(i, j) \in A$. The objective is to minimize the average total cost per unit time,

$$c(T) = \sum_j g_j T_j + K_j / T_j.$$

Roundy (1985) extended the Maxwell-Muckstadt model by considering joint setup costs and relaxing the nestedness condition. He showed that the total cost for this case is

$$c(T) = \sum_R g_R \max\{T_j\colon j \in R\} + \sum_F K_F \Big/ (\min\{T_j\colon j \in F\}),$$

where R and F are suitably defined subsets of products, and g_R and K_F are corresponding holding and production costs. Although the constraints $T_i \le T_j$ thus disappear, the modeling capabilities of variable upper-bound constraints are reflected in the handling of joint setup costs and holding costs. For that, Roundy extended the product set N by adding the R and F sets, and "defined" corresponding variables T_R (T_F, respectively) by the inequalities $T_j \le T_R$ ($T_F \le T_j$, resp.) for all $j \in R$ (F, resp.). The resulting problem is thus cast again into the Maxwell-Muckstadt form above. Roundy's major result is that *optimal power-of-two policies* thus constructed are *94% effective*; that is, the cost of an optimal policy cannot be less than 94% that of an optimum power-of-two policy. He also showed that searching for an optimal base interval T_0 yields a *98%-effective* solution. The introduction of CCC extends this approach to general convex average cost functions $f_j(k_j) = c_j(T_0 2^{k_j})$. The 94% and 98% effectiveness results, however, hold only for the specific functions $c(\,)$ above.

3.2.2. Bayesian Estimation Subject to Rank Order Constraints. Statistical problems of *partially ordered estimation* have been discussed extensively in the literature, (see, e.g., Veinott 1971 and Barlow et al. 1972). Let $p_1, \ldots, p_n$ denote parameters to be jointly estimated and let $f_j(x_j)$ denote the *loss* associated with estimating that $p_j = x_j$ for $j = 1, \ldots, n$. The model being estimated may specify a *partial order* on the parameters, as reflected by constraints $x_j \le x_i$ for a set A of pairs (i, j), as well as simple upper and lower bounds on the parameter values. If, in addition, the model requires the parameter values to be integer, then the problem of jointly estimating the parameter values to minimize total loss is precisely an instance of problem CCC. If there is no such integrality restriction, then the problem is an instance of the continuous relaxation of CCC.

The algorithm of Hochbaum and Queyranne (2003) solves CCC with complexity $O(mn \log(n^2/m) + n \log U)$. In Bayesian estimation the functions are typically quadratic where each term is of the form $((x_i - \mu)/\sigma)^2$ for μ and σ the mean and the standard deviation of the distribution of the ith random variable. For quadratic convex functions the algorithm runs in strongly polynomial time $O(mn \log(n^2/m) + n \log n)$. For the isotonic regression problem, where the order is linear, the running time improves to $O(n \log n + n \log U)$, and thus the complexity of the algorithm is $O(n \log(\max\{n, U\}))$. Ahuja and Orlin (2001) reported on a different $O(n \log U)$ time algorithm for isotonic regression. The Bayesian estimation problem has been researched extensively in the statistical study of observations. The book by Barlow et al. (1972) provides an excellent review of applications and algorithms for Bayesian estimation subject to rank order constraints.

3.2.3. Medical Prognosis. Medical prognosis involves estimates of cure, complication, recurrence of disease, length of stay in health-care facilities, or survival for a patient or group of patients. Accurate assessment of a patient's prognosis is essential for determining an appropriate medical treatment plan.

RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

A wide range of techniques has been used for prognosis estimation varying from Markov chain techniques, to regression, to linear programming, to genetic algorithms and neural networks.

The process of assessing medical prognosis for patients suffering from diseases such as cardiovascular illness, stroke, or various types of cancer relies on past data and medical research generating knowledge about the correlation between certain medical measurements and the prognosis. This is a classical Bayesian estimation scenario. There are initial predictions and prognosis estimates for specific *cases* represented by arrays of medical measurements of relevant parameters. There are observations of patient data that have a partial ordering based on medical knowledge and experience. A patient with certain measurements is expected to have longer life span before recurrence than another patient with a different set of measurements. This is represented as ordering between pairs of cases indicating that one has better prognosis than the other without specifically quantifying it. There is a penalty for making an error in overestimating and underestimating the time until recurrence.

All medical prognosis methods use databases to capture existing medical knowledge in the form of cases represented by arrays of values of medical parameters and estimated prognoses, for example, in the form of time to recurrence in the case of heart attack. Past experience also includes the relative rank ordering between pairs of some cases indicating that the prognosis for one is better than the prognosis for the second. The estimates for known cases are given with a certain confidence level. Existing techniques focus on obtaining a prognosis in the form of binary outcome indicating whether the prognosis is *good* or *bad*.

Ryu et al. (2004) recently extended the range of capabilities of medical prognosis by developing estimates that provide a time interval (to next recurrence) rather than just a binary outcome. In their model they include penalties for existing estimates in the database. The penalties are in the form of a linear cost for deviating above or below the estimate forming a convex piecewise linear function (with two pieces each). With the linear penalties assumption and allowing for partial order information—referred to as *monotonicity*—the authors established that the linear programming problem modeling the problem of finding adjusted estimates to minimize the deviation penalties can be solved efficiently with a minimum-cost network flow model.

The medical prognosis estimate problem fits ideally into the CCC model. Each known case in the database is represented as a node in the graph with a convex penalty function associated with deviating from the expected prognosis estimate. There is a partial order between different cases represented by arcs in the graph that indicate pairwise comparison (i, j) implying that a certain array of parameter readings of case i leads to more favorable prognosis than the one for case j. Allowing for general convex functions as penalties permits a refinement of the representation of the degree of confidence in past estimates. Moreover, the representation as a CCC problem and the implied algorithm based on minimum cut is significantly more efficient than the algorithm proposed for the simpler linear penalties case.

The model can be further refined to allow ordering that is known with uncertainty. When a violation of the ordering is linear in the difference between the estimates, that model is still solved with the same complexity as CCC as a convex s-excess problem. If the violation is arbitrary with convex costs for violation, the problem is then a convex DMCNF.

3.2.4. Firm Bankruptcy Risk Analysis. Assessing a firm's bankruptcy risk has long been an issue of major concern to the financial and accounting communities. A typical prediction model involves first identifying a list of parameters that are deemed relevant to corporate bankruptcy. Based on financial and accounting knowledge and past data experience, there are estimates of the bankruptcy risk of a firm characterized by a given array of parameter values. There is a partial order between pairs of arrays that correspond to the assessment that the risk of bankruptcy of a firm with one set of parameter values is higher than that with another set. This ordering utilizes past data and knowledge on correlations and causal patterns between parameter values and scenarios and bankruptcy risk.

To assess whether a firm is at risk of bankruptcy, a common practice is to develop a linear model based on regression analysis. This model is used along with a *threshold value*. Plugging in the values of the parameters for a given corporation results in a real value that is compared to the threshold value. If the value is smaller than the threshold, then the risk of defaulting is high; otherwise it is low. Hence these models provide only binary information.

There are numerous examples of bankruptcy analysis that are of the linear regression type. One well-known classical model is by Altman (1968). Altman was also the first to successfully use stepwise multiple discriminate analysis to develop a prediction model with a high degree of accuracy. Using a sample of 66 companies in which 33 failed and 33 were successful, Altman's model achieved an accuracy rate of 95.0%. Altman's model uses the parameters

$A =$ working capital/total assets
$B =$ retained earnings/total assets

RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

$C =$ earnings before interest and taxes/total assets
$D =$ market value of equity/book value of total debt, and
$E =$ sales/total assets.

The linear model takes the following form:

$$Z = 1.2A + 1.4B + 3.3C + 0.6D + 0.999E.$$

For $Z < 2.675$—the threshold value—the firm is classified as "failed."

Another successful predictive model is the *CA-Score model*. The threshold value in that model is determined so that if CA-Score < -0.3 then the firm is classified as "failed." The CA-Score model is reported in Legault (1987) to have an average reliability rate of 83% and is restricted to evaluating manufacturing companies.

An optimization model based on CCC allows for finer, and potentially more accurate, assessment of the risk of failure and bankruptcy. Using the historical data base we associate with each firm's array of parameter values (a node) the length of time that has elapsed since the reading of these parameter values until bankruptcy. For firms that have not defaulted, an estimated time to default can be introduced. Or, if the estimate is that the firm is not to default, we set a large value (simulating infinity) as the length of time to defaulting. Using financial knowledge expertise, there is a partial order (arcs) between data arrays indicating which are more likely to default before others.

When assessing or correcting estimates of *time to bankruptcy*, we define a graph in which the nodes correspond to firms or hypothetical arrays of parameter values, and the estimated time to bankruptcy for each. There are penalties in the form of convex function for the cost of deviating from the given estimates of time to bankruptcy. Arcs in the graph correspond to the partial order. The problem of obtaining revised estimates that minimize the total penalties is a convex closure problem.

Solving this CCC problem provides information on the estimated time to bankruptcy for each new data point and may revise the existing estimates in the database. This information is more refined than the binary yes/no threshold information. Hence, the model can then be used as a predictive tool as well as to revise the estimates in the database and improve the quality of the historical information as more data becomes available.

Refining the model by allowing us to violate the ordering at additional penalty is analogous to the refinement of the medical prognosis model.

3.3. 2-Approximation Algorithms for Integer Programming with Two Variables per Inequality

Many optimization problems of scheduling, location, resource allocation, capacity expansion, lot-sizing, and other applications of primary concern to management are NP-hard. There are no efficient algorithms known for NP-hard problems, and it is widely believed that no efficient algorithms exist. There is nevertheless a need to solve such problems, and thus a practical approach is to use heuristics that work efficiently.

The quality of the solutions (or the size of the error) delivered by a heuristic is naturally of concern. In the analysis of approximation algorithms the goal is to find among all efficient algorithms that provide feasible solutions to a problem those that minimize the *worst-case error*. The worst-case error, also known as *approximation ratio*, is the largest ratio of the solution value divided by the optimum across all possible problem instances. So a 2-approximation algorithm guarantees that for any instance of the problem the value of the solution delivered is, at most, twice the optimum. In practice, the error observed is much lower than the worst-case error bound. It has been observed in practice that an algorithm with a smaller approximation ratio tends to deliver better solutions. This motivated the search for low ratio approximation algorithms. But the ad-hoc nature of this search makes the derivation of results problem specific and technically involved (e.g., many recent approximation algorithms depend on the use of the ellipsoid method). Details on the background and current state of research in approximation algorithms are available in Hochbaum (1997).

It is therefore significant that, for a wide class of problems, one can generate, immediately from the formulation, 2-approximation algorithms using the closure problem. This class of problems are those that can be formulated as IP2.

IP2 is an integer programming problem with each constraint having at most two variables. Unlike the monotone inequalities problem (IPM), the signs of the coefficients are unrestricted. Problems with two variables per inequality are commonplace. Major problem categories include the vertex cover problem, the independent set problem, a variant of the maximum-clique problem, several types of satisfiability problems, and others. The problem formulation is

$$\text{(IP2)} \qquad \min \quad \sum_{j=1}^{n} w_j x_j$$

$$\text{subject to} \quad a_i x_{j_i} + b_i x_{k_i} \geq c_i \quad (i = 1, \ldots, m)$$

$$l_j \leq x_j \leq u_j, \quad x_j \text{ integer } (j = 1, \ldots, n),$$

RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

Problem IP2 is in general NP-hard because the fundamental vertex cover problem is a special case of IP2 with constraints on binary variables of the type $x_i + x_j \geq 1$. The vertex cover problem was among the first problems to be proved NP-hard by Karp (1972).

For the vertex cover problem, the ratio of 2 is the best (smallest) approximation ratio known, and accumulating evidence points to the possibility that no better approximation ratio can be obtained for the vertex cover problem by a polynomial time algorithm. It is therefore of interest that the vast set of problems IP2 generalizing vertex cover all have polynomial time (efficient) 2-approximation algorithms, all generated by solving a certain minimum-closure problem Hochbaum et al. (1993). The main idea of the algorithm is to *reduce* an IP2 problem to a monotone problem IPM. We give a sketch of this idea next.

Consider a generic nonmonotone inequality of the form $ax + by \geq c$ where a and b are of the same sign. Each variable x is replaced by two variables, x^+ and x^-, and each inequality by two monotone inequalities:

$$ax^+ - by^- \geq c$$
$$-ax^- + by^+ \geq c.$$

Any feasible solution to the two inequalities satisfy $a(x^+ - x^-) + b(y^+ - y^-) \geq 2c$. The upper and lower bounds constraints $l_j \leq x_j \leq u_j$ are transformed to

$$l_j \leq x_j^+ \leq u_j$$
$$-u_j \leq x_j^- \leq -l_j.$$

In the objective function, the variable x is substituted by $\frac{1}{2}(x^+ - x^-)$. Monotone inequalities remain so by replacing the variables x and y in one inequality by x^+ and y^+, and in the second, by x^- and y^-, respectively. Once the transformation is complete, the integer programming on monotone inequalities can be solved in integers as a minimum-closure problem. The solution is mapped back to IP2 by the substitution above that may create a half-integral solution. Hochbaum et al. (1993) showed that for feasible IP2, the half-integral solution can be rounded to an integer solution that is a 2-approximate solution.

The complexity of the algorithm is dominated by the complexity of the procedure in Hochbaum and Naor (1994) for optimizing over a monotone system. The running time is $O(mnU^2 \log(Un^2/m))$ for $U = \max_{j=1,\ldots,n}\{u_j - l_j\}$.

3.4. Approximation Algorithms for 2var Problems

Like IP2, 2var problems can be approximated by a process of "monotonizing" and "binarizing" the formulation and solving the resulting formulation as an s-excess problem. Mapping back the solution to the original problem yields a half-integral solution. If a feasible rounding can be found, then that rounded integer solution is a 2-approximation. Details on the procedure and a list of applications are provided in Hochbaum (2002).

3.5. Applications of 2var

We sketch here two applications of 2var problems.

3.5.1. Forestry. The Generalized Independent Set problem is a 2var problem generalizing the well-known independent set problem. In the independent set problem we seek a set of nodes of maximum total weight so that no two are adjacent. In the Generalized Independent Set problem it is permitted to have adjacent nodes in the set, but at a penalty that may be positive or negative. The independent set problem is a special case of Generalized Independent Set where the penalties are infinite. The formulation of the Generalized Independent Set problem is

$$\begin{aligned} \text{(Gen-Ind-Set)} \quad \max \quad & \sum_{j\in V} w_j x_j - \sum_{(i,j)\in E} c_{ij} z_{ij} \\ \text{subject to} \quad & x_i + x_j - z_{ij} \leq 1 \quad (i,j) \in E \\ & x_i,\ z_{ij} \quad \text{binary for all } i,j. \end{aligned}$$

The Generalized Independent Set problem was introduced by Hochbaum and Pathria (1997) as a model of two forest harvesting optimization problems. The first problem assigns benefits for harvesting forest cells and penalties for harvesting *adjacent* cells; the second problem assigns benefits for harvesting cells as well as benefits for creating borders that separate harvested and unharvested cells. The objective is to identify the set of cells to harvest to maximize the net benefits.

Although not immediately apparent, these problems were shown in Hochbaum and Pathria (1997) to be equivalent to the Generalized Independent Set problem on a graph $G = (V, E)$ with node weights and edge weights. Approximation algorithms with a worst-case ratio of 2 are then immediately implied.

The cell arrangement in forests is often gridlike. The corresponding graph in that case is bipartite and the problem is then a *monotone* 2var. The problems on gridlike forest are then solved in polynomial time as a closure problem.

3.5.2. Location of Competing Facilities. Another application of the Generalized Independent Set problem is the problem of locating postal services (Ball 1992). Each potential location of the service has a utility value associated with it. The value, however, is diminished when several geographically proximal facilities compete for customers. Following the principle of inclusion-exclusion, the second-order approximation of that loss is represented by pairwise interaction cost for every pair of potential facilities.

RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

The postal service problem is defined on a complete graph $G=(V,E)$ where the pairwise interaction cost, c_{ij}, is assigned to every respective edge (i,j). Since Generalized Independent Set is a 2var problem, half-integral solutions are immediately available by solving the appropriate minimum-cut problem. Furthermore, when the underlying graph for Generalized Independent Set is bipartite then the problem is monotone 2var and solvable in polynomial time (Hochbaum and Pathria 1997). The location arrangements can often be bipartite if they correspond, for instance, to a grid street arrangement.

3.6. Applications of Convex *s*-Excess

3.6.1. Capacity Expansion. Capacity expansion decisions are some of the most critical decisions made by management, as the capacity level and its technological features affect capital investment and the capability and quality of satisfying demand. Machine costs have been rising over time with improved and sophisticated technological progress and therefore the decision to invest typically involves a capital outlay that can affect the financial position of the firm. When a specific machine model becomes available for acquisition, its purchase can improve the competitiveness of the firm. On the other hand, the price of the model is likely to decrease over time as more advanced technology becomes available, thus delaying the acquisition can reduce costs.

Three recent papers have studied the capacity expansion problem: Çakaniyildirim et al. (2004), Zhang et al. (2004), and Huh and Roundy (2002). In these papers it is assumed that capacity cost is a decreasing function over time, and a delay in acquisition may result in lost sales if capacity is insufficient to meet demand. With convex functions for lost-sales penalties, balancing these requirements, as well as potential inventory issues, is a CCC or convex *s*-excess problem.

Each point in time and a certain capacity level is represented as a node in the graph. There are arcs between earlier time periods and later time periods as well as between capacity selections and others containing and complementing them. With each node there is a lost-sale convex penalty function. Each arc going between two nodes representing different points in time and different capacities has the cost of expanding the capacity before the later period and potential cost of inventory. The problem of minimizing the cost of lost sales, cost of purchase, and costs of inventory is a convex *s*-excess problem. With convex costs on the arcs the problem is convex DMCNF.

3.6.2. The Image-Segmentation and Error Correction. In the problem of image segmentation an image is transmitted and degraded by noise. The goal is to reset the values of the colors to the pixels to minimize the penalty for the deviation from the observed colors and, furthermore, so that the discontinuity in terms of separation of colors between adjacent pixels is as small as possible.

Consider an image consisting of a set of pixels each with a given color and a neighborhood relation between pairs of pixels. In the image-segmentation problem, each pixel gets a color assignment that may be different from the given color of the pixel so that neighboring pixels will tend to have the same color assignment. The aim is to modify the given color values as little as possible while penalizing changes in color between neighboring pixels. The penalty function thus has two components: the deviation cost that accounts for modifying the color assignment of each pixel, and the separation cost that penalizes pairwise discontinuities in color assignment for each pair of neighboring pixels.

Representing the image-segmentation problem as a graph problem, we let the pixels be nodes in a graph and the pairwise neighborhood relation be indicated by edges between neighboring pixels. Each pairwise adjacency relation $\{i,j\}$ is replaced by a pair of two opposing arcs (i,j), and (j,i), each carrying a capacity representing the penalty function for the case that the color of j is greater than the color of i and vice versa. The set of directed arcs representing the adjacency (or neighborhood) relation is denoted by A. We denote the set of neighbors of i, or those nodes that have pairwise relation with i, by $N(i)$. Thus the problem is defined on a graph $G=(V,A)$.

Let each node j have a value g_j associated with it—the observed color. The problem is to assign an integer value x_j to each node j to minimize the penalty function. Let the K color shades be a set of ordered values $\mathcal{L}=\{q_1,q_2,\ldots,q_K\}$. Denote the assignment of a color q_p to pixel j by setting the variable $x_j=p$. Each pixel j is permitted to be assigned any color in a specified range $\{q_{l_j},\ldots,q_{u_j}\}$. For $G()$ the *deviation-cost* function and $F()$ the *separation-cost* function the problem is,

$$\min_{u_i\ge x_i\ge l_i} \sum_{i\in V} G_i(g_i,x_i)+\sum_{i\in V}\sum_{j\in N(i)} F_{ij}(x_i-x_j).$$

This formulation is equivalent to the following constrained optimization problem, referred to as IS (for image segmentation):

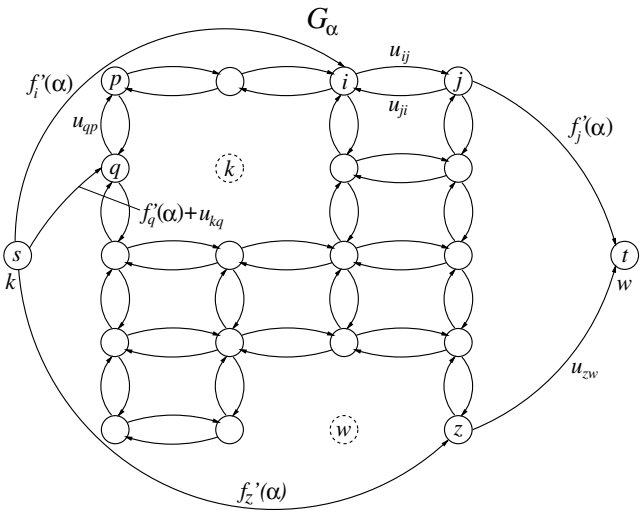
$$\begin{aligned}
\text{(IS)}\qquad \min\quad & \sum_{j\in V} G_j(g_j,x_j)+\sum_{(i,j)\in A} F_{ij}(z_{ij})\\
\text{subject to}\quad & x_i-x_j\le z_{ij} \quad \text{for } (i,j)\in A\\
& u_j\ge x_j\ge l_j \qquad j=1,\ldots,n\\
& z_{ij}\ge 0 \qquad (i,j)\in A.
\end{aligned}$$

Notice that the refinements of the medical diagnosis problem and the time-to-bankruptcy problem as well

RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

Figure 4 **The Graph G_α**



as the capacity expansion problem are all problems formulated as IS.

The constraints of IS identify it as an *s*-excess problem. Depending on the objective function, the algorithms will be either based on parametric minimum cut as for the convex *s*-excess problem, or on multiple calls for a minimum-cut procedure as for the convex DMCNF. For $G(\,)$ convex functions and $F(\,)$ linear functions the problem is the convex *s*-excess. If $F(\,)$ are convex as well then the problem is convex DMCNF (Ahuja 2003).

The convex *s*-excess problem is solved in this case as a minimum-cut problem on a graph parametrized by α where source and sink adjacent arc capacities are the derivatives of the respective node functions $G_j(\,)$ at α. An example of such a graph for a two-dimensional image is given in Figure 4. In this example, nodes k and w are "shrunk" with source and sink respectively accounting for previous iterations where it was established that $k \le \alpha$ and $w > \alpha$.

3.7. Applications of the Convex Dual of Minimum-Cost Network Flow

A number of additional applications beyond those presented here are described in Ahuja et al. (2003).

3.7.1. The Convex Penalty Image Segmentation. When the functions $F(\,)$ and $G(\,)$ are both convex then the IS problem is cast as a convex DMCNF. The algorithms of Ahuja et al. (2003, 2004) solve this problem efficiently. The algorithm of Ahuja et al. (2004) is based on reduction to the *s*-excess problem, and it works also, at larger complexity, when the functions $G(\,)$ are nonlinear (rather than convex). The running time in that case depends on the largest interval length U in which each variable is restricted and the algorithm is thus pseudopolynomial.

3.7.2. Scheduling in Project Management. Scheduling in project management attempts to eliminate waste by reducing slack times. This scheduling application has been adapted by Ahuja et al. (2003) from Levner and Nemirovsky (1991). We denote a project by a directed graph $G = (V, A)$, where set A denotes jobs, and the node set V denotes events (known as the Activity-on-Arc model). The network G also captures precedence relations among the arcs. The completion times of all jobs are assumed to be fixed. Let t_{ij} denote the time it takes to complete job (i, j). (Notice that t_{ij} is not a decision variable in this problem.) Let ν_i denote the time for event i.

Consider the feasible event times, i.e., ν_is satisfying $\nu_j - \nu_i \ge t_{ij}$ for all $(i, j) \in A$. With respect to these event times, a job (i, j) will be completed at time $\nu_i + t_{ij}$ but the jobs emanating from node j will start at time ν_j. Let $w_{ij} = \nu_j - \nu_i - t_{ij}$ denote the slack time, and let $F_{ij}(w_{ij})$ denote its associated penalty cost. This penalty cost may capture the lost-opportunity cost of the capital tied up or some other factors (such as perishability or deterioration in quality) that make slack times undesirable. There may also be some upper bounds β_{ij} on slack times. We may assume without loss of generality that the lower bound on slack time is 0, since any other lower bound would be incorporated into the times to complete a task. The project-scheduling problem is to obtain event times ν_is so that the project is completed within the specified time period T and the penalty cost associated with job slacks is minimal. With this notation, the problem formulation is,

$$
\begin{aligned}
\text{(Schedule-PM)} \qquad \min \quad & \sum_{(i,j)\in A} F_{ij}(w_{ij}) \\
\text{subject to} \quad & \nu_t - \nu_s \le T \\
& \nu_j - \nu_i = w_{ij} + t_{ij} \\
& \qquad \text{for all } (i, j) \in A \\
& 0 \le w_{ij} \le \beta_{ij} \\
& \qquad \text{for all } (i, j) \in A
\end{aligned}
$$

Because this problem is convex DMCNF, it can be solved by reducing it to a sequence of minimum-cut problems as described in Ahuja et al. (2004).

4. Conclusions

This survey is testimony to the broad and deep impact of three papers on optimization subjects that appeared in *Management Science* in the 1970s. The influence of that research has extended far beyond the motivating applications to theory and applications that could not have been envisioned at the time the research was conducted. This underscores the long-lasting value of truly good research that frames

RIGHTSLINK

Downloaded from informs.org by [128.32.10.230] on 14 May 2018, at 01:26 . For personal use only, all rights reserved.

a problem in a generic way so that solutions and methodology have power far beyond the application at hand. It also illustrates the cumulative nature of the research process by showing how solution of an elementary problem serves as a stepping stone to solution of much more complex problems.

Finally, looking back with pride on the impact of old optimization work published in *Management Science* makes us appreciate more deeply the need to publish good optimization papers today. In addition to addressing important contemporary management problems, who knows what vital issues of the future these works may impact?

Acknowledgments

Research supported in part by NSF awards DMI-0085690 and DMI-0084857.

References

Ahuja, R. K., J. B. Orlin. 2001. A fast scaling algorithm for minimizing separable convex functions subject to chain constraints. *Oper. Res.* **49**(5) 784–789.

Ahuja, R. K., D. S. Hochbaum, J. B. Orlin. 2003. Solving the convex cost integer dual network flow problem. *Management Sci.* **49**(7) 950–964.

Ahuja, R. K., D. S. Hochbaum, J. B. Orlin. 2004. A cut based algorithm for the convex dual of the minimum cost network flow problem. *Algorithmica.* Forthcoming.

Altman, E. I. 1968. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *J. Finance* **23** 589–609.

Balinski, M. L. 1970. On a selection problem. *Management Sci.* **17**(3) 230–231.

Ball, M. 1992. Locating competitive new facilities in the presence of existing facilities. *Proc. 5th United States Postal Service Adv. Tech. Conf.*, 1169–1177.

Barlow, R. E., D. J. Bartholomew, J. M. Bremer, H. D. Brunk. 1972. *Statistical Inference Under Order Restrictions.* Wiley, New York.

Çakaniyildirim, M., R. O. Roundy, S. C. Wood. 2004. Optimal machine capacity expansions with nested limitations under demand uncertainty. *Naval Res. Logist.* Forthcoming.

Gale, D. 1957. A theorem of flows in networks. *Pacific J. Math.* **7** 1073–1082.

Gallo, G., M. D. Grigoriadis, R. E. Tarjan. 1989. A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.* **18**(1) 30–55.

Goldberg, A. V., R. E. Tarjan. 1988. A new approach to the maximum flow problem. *J. ACM* **35** 921–940.

Hochbaum, D. S. 1994. Lower and upper bounds for allocation problems. *Math. Oper. Res.* **19**(2) 390–409.

Hochbaum, D. S. 1997a. *Approximation Algorithms for NP-Hard Problems.* PWS, Boston, MA.

Hochbaum, D. S. 1997b. The pseudoflow algorithm for the maximum flow problem. Manuscript revised 2002, University of California, Berkeley, CA.

Hochbaum, D. S. 2001a. A new-old algorithm for minimum-cut and maximum-flow in closure graphs. 30th Anniversary Special Paper. *Networks* **37**(4) 171–193.

Hochbaum, D. S. 2001b. An efficient algorithm for image segmentation, Markov random fields and related problems. *J. ACM* **48**(4) 686–701.

Hochbaum, D. S. 2002. Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations. *Eur. J. Oper. Res.* **140**(2) 291–321.

Hochbaum, D. S. 2003. Efficient algorithms for the inverse spanning tree problem. *Oper. Res.* **51**(5) 785–797.

Hochbaum, D. S., J. Naor. 1994. Simple and fast algorithms for linear and integer programs with two variables per inequality. *SIAM J. Comput.* **23**(6) 1179–1192.

Hochbaum, D. S., A. Pathria. 1997. Forest harvesting and minimum cuts. *Forest Sci.* **43**(4) 544–554.

Hochbaum, D. S., M. Queyranne. 2003. The convex cost closure problem. *SIAM J. Discrete Math.* **16**(2) 192–207.

Hochbaum, D. S., J. G. Shanthikumar. 1990. Convex separable optimization is not much harder than linear optimization. *J. ACM* **37** 843–862.

Hochbaum, D. S., N. Megiddo, J. Naor, A. Tamir. 1993. Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality. *Math. Programming* **62** 69–83.

Hoffman, A. J. 1960. Some recent applications of the theory of linear inequalities to extremal combinatorial analysis. R. Bellman, M. Hall Jr., eds. *Proc. Sympos. Appl. Math.* Vol. X. *Combinatorial Anal.*, American Mathematical Society, Providence, RI, 113–127.

Huh, W. T., R. O. Roundy. 2002. A continuous-time strategic capacity planning model. Working paper, Cornell University, Ithaca, NY.

Johnson, T. B. 1968. Optimum open pit mine production scheduling. Ph.D. thesis, Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA.

Karp, R. M. 1972. Reducibility among combinatorial problems. R. E. Miller, J. W. Tatcher, eds. *Complexity of Computer Computations.* Plenum Press, NY, 85–103.

Lawler, E. L. 1976. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart and Winston, New York.

Legault, J. 1987. C.A.—Score, a warning system for small business failures Bilanas. *Insolvency Prediction*. E. Sands & Associates, Inc., www.sands-trustee.com/insolart.htm.

Lerchs, H., I. F. Grossmann. 1965. Optimum design of open-pit mines. *Transactions, C.I.M.* **LXVIII** 17–24.

Levner, E. V., A. S. Nemirovsky. 1991. A network flow algorithm for just-in-time project scheduling. Memorandum COSOR 91-21, Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands.

Mamer, J. W., S. Smith. 1982. Optimizing field repair kits based on job completion rate. *Management Sci.* **28**(11) 1328–1333.

Maxwell, W. L., J. A Muckstadt. 1985. Establishing consistent and realistic reorder intervals in production/distribution systems. *Oper. Res.* **33**(6) 1316–1341.

Picard, J. C. 1976. Maximal closure of a graph and applications to combinatorial problems. *Management Sci.* **22** 1268–1272.

Radzik, T. 1993. Parametric flows, weighted means of cuts, and fractional combinatorial optimization. P. M. Pardalos, ed. *Complexity in Numerical Optimization.* World Scientific, Singapore, 351–386.

Rhys, J. M. W. 1970. A selection problem of shared fixed costs and network flows. *Management Sci.* **17**(3) 200–207.

Roundy, R. 1985. A 98%-effective integer-ratio lot-sizing for one-warehouse multi-retailer systems. *Management Sci.* **31** 1416–1430.

Ryu, Y. U., R. Chandrasekaran, V. Jacob. 2004. Disease prognosis with an isotonic prediction technique. *Management Sci.* **50**(6) 777–785.

Veinott, A. F., Jr. 1971. Least *d*-majorized network flows with inventory and statistical applications. *Management Sci.* **17** 547–567.

Zhang, F., R. O. Roundy, M. Çakanyildirim, W. T. Huh. 2004. Optimal capacity expansion for multi-product, multi-machine manufacturing systems with stochastic demand. *IIE Trans.* **36** 23–36.

RIGHTSLINK