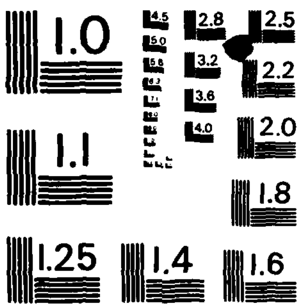
AD-A137 773 HIDDEN AND EMBEDDED STRUCTURE IN LINEAR PROGRAMS(U) 1/1
NORTHWESTERN UNIV EVANSTON IL DEPT OF INDUSTRIAL
ENGINEERING AND MANAGEMENT SCIENCES R E BIXBY
UNCLASSIFIED 30 SEP 83 AFOSR-TR-84-0046 AFOSR-82-0004 F/G 9/2 NL

END
DATE
FILMED
3 84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFOSR-TR- 84-0046 (2) AFOSR-82-0004

FINAL REPORT: AFOSR Grant [redacted]

"Hidden and Embedded Structure in Linear Programs"

Principal Investigator: Robert E. Bixby

AFOSR-82-0004

This report ~~The~~ following is a summary of work completed on AFOSR Grant ~~82-0004.~~ The summary begins with a listing of papers written followed by further descriptions of work completed as well as work in progress. Contents include:

I. PAPERS WRITTEN

[1] A Simple Theorem on 3-Connectivity. Linear Algebra and Its Applications 45 123-126.

[2] A Compositio for Perfect Graphs. To appear in a volume edited by V. Chvátal and C. Berge.

[3] The Partial Order of a Polymatroid Extreme Point; (with W.H. Cunningham and D.M. Topkis). To appear in Mathematics of Operations Research.

[4] Algorithms for Two Versions of Graph Realization and an Application to Linear Programming (1983). To appear in the Proceedings of the Waterloo Silver Jubilee Conference.

[5] A Note on Recognizing Path Matrices; and

[6] Packing and Covering by Integral Feasible Flows in Integral Supply-Demand Networks, (with O. M.-C. Marcotte and L.E. Trotter, Jr.)

II. THESES SUPERVISED

Donald K. Wagner, Assistant Professor at Purdue University. Dissertation title: An Almost Linear-Time Algorithm for Graph Realization.

III. WORK COMPLETED AND IN PROGRESS

A. Polymatroids and Matroid Intersection

The work described in [3] was presented at the XI International Mathematical Programming Symposium, Bonn, West Germany, at the Oberwolfach Workshop on Mathematical Programming, held at Oberwolfach, West Germany, and at the Bielefeld meeting on Applications of Matroids, held at the University of Bieliefeld, West Germany. Efforts to extend the results of [3] are continuing.

The problem motivating [3] may be described as follows. Let E be a finite set, and let f be a real-valued function defined on subsets of E. f is said to be <u>submodular</u> if

$$f(A) + f(B) \geqslant f(A \cup B) + f(A \cap B)$$

for each pair of subsets $A, B \subseteq E$. Submodular functions play a central role in the study of combinatorial optimization problems, not unlike the role played by convex funtions in continuious optimization. A good discussion along these lines is given in "Submodular Functions and Convexity" by L. Lovász (<u>Mathematical Programming: The State of the Art</u>, edited by A. Bachem, M. Grötschel and B. Korte, Bonn 1982). The major open problem in this theory is that of finding a direct combinatorial procedure for minimizing submodular functions. Algorithms have been found for several special cases, and it is hoped that the methods developed in [3] will lead to a solution in the general case. An algorithm is given in [3], and we have recently succeeded in showing that this algorithm is finite. A very short proof has also been found for a result of Topkis, characterizing adjacency in polymatroids.

A problem related to submodular function minimization is the so-called weighted matroid intersection problem. Collette Coullard, a Ph.D. Student at Northwestern University, is working to simplify various rewults in this area (including Lawler's presentation of his primal matroid intersection algorithm), and on developing a simplex method for matroid intersection. It is also hoped that her work may lead to an appropriate definition for "weighted matroid partition," and thus to a better explanation of the duality phenomena apparent in these problems. The following proposed generalization has grown out of her work. A <u>matroid</u> M on a finite set E is a collection I of subsets of E, called <u>independent sets</u>, such that the empty set is independent, every subset of an independent set is independent, and for every $A \subseteq E$, every maximal independent subset of A has the same size. Let $M_1, \ldots, M_k$ be matroids on the same underlying set E. Then it is well known that the classical matroid greedy algorithm can be used to maximize a linear functional on the matroid <u>sum</u> $M_1 + \ldots + M_k$. Where I is a subset of E, I is independent in this sum matroid if and only if it can be partitioned into sets $I_1, \ldots, I_k$ independent in the matroids $M_1, \ldots, M_k$, respectively. Suppose, now, that instead of a single linear functional, we take a separate functional for each of the matroids, and then ask for the maximum weight partitionable subset of elements, where the weight applied to a particular element is determined by the independent set I_j that contains it in the partition. This problem can be shown to be equivalent to a restricted cardinality version of matroid intersection, and it may be possible, in turn, to reduce this to the classical weighted intersection problem. Conversely, it is easy to see that weighted intersection can, indeed, be reduced to this new weighted partition problem.

B. Algorithms for Recognizing Hidden and Embedded Structures in Linear Programming Problems.

Papers [4] and [5] are connected with this work. Paper [4] was presented as an invited address at the Waterloo Silver Jubilee Conference in Combinatorics, held at the University of Waterloo (Waterloo Ontario, Canada) in June 1982.

Work on this problem has proceeded in several directions:

1. As noted in an earlier report on this grant, one of the major tasks that had to be undertaken in this work, before computational testing on hidden networks could begin, was to convert our existing codes to a system allowing greater storage capabilities so that testing can be carried out on larger linear programs (the previous code was not capable of handling problems with more than approximately 200 rows).

2. Here we describe in some detail our computational experience with embedded and hidden networks in 10 test LPs (though not all LPs have been used in all experiments). Given a LP is equality form, we say this LP has a hidden network if some subset of its constraints can be discarded yielding a problem that, by elementary row operations, can be converted to a linear network flow problem. If these row operations involve only row and column scaling, then the result is called an embedded network. The initial statistics for the constraint matrices of the test LPs were as follows:

LP name	Rows	Columns	Nonzero Elements	Density
ETA	335	704	2231	0.95%
GIF-PIN	617	1092	3467	0.51%
RECIPE	98	180	768	4.35%
SCAGR7	130	140	553	3.04%
SCFXM1	331	457	2612	1.73%
SCSD1	78	760	3148	5.31%
SCTAP1	311	580	3372	1.87%
SC205	206	203	552	1.32%
STAIR	357	467	3857	2.31%
VTPBASE	199	203	914	2.26%

These LPs were first analyzed to remove certain inessential rows and columns. Free rows, fixed columns, and all-zero rows and columns of the LP were deleted initially. Rows that had the effect of placing lower or upper bounds on variables were then removed, since efficient simplex codes need not handle bounds as explicit constraints.

Unit columns were then added to represent slack variables on any inequality constraints. Statistics for the revised constraint matrices were as follows:

LP name	Rows	Columns	Nonzero Elements	Density
ETA	275	699	1848	0.96%
GIF-PIN	590	1150	2409	0.36%
RECIPE	82	174	644	4.51%
SCAGR7	89	144	378	2.95%
SCFXM1	289	559	2641	1.63%
SCSD1	77	760	2388	4.08%
SCTAP1	300	760	1872	0.82%
SC205	203	315	663	1.04%
STAIR	356	532	3813	2.01%
VTPBASE	166	296	880	1.79%

We used a scaling heuristic to try to increase the number of rows that contained only one nonzero magnitude; these rows are trivially scalable to ±1 rows. The following table shows the number of such rows, before scaling and after the best scaling:

LP name	Rows	Single-Magnitude Rows: Unscaled		Single-Magnitude Rows: Scaled	
ETA	275	31	(11%)	155	(56%)
GIF-PIN	590	264	(45%)	590	(100%)
RECIPE	82	49	(60%)	53	(65%)
SCAGR7	89	46	(52%)	63	(71%)
SCFXM1	289	88	(30%)	129	(45%)
SCSD1	77	0	(0%)	39	(51%)
SCTAP1	300	120	(40%)	126	(42%)
SC205	203	145	(71%)	147	(72%)
STAIR	356	64	(18%)	165	(46%)
VTPBASE	166	22	(13%)	63	(38%)

We then applied extraction heuristics to find embedded networks with the ±1 rows. Best results are shown below. The largest network matrix that could be extracted from the unscaled matrix is also given, to indicate the usefuleness of scaling.

LP name	Rows	+-1 Rows		Single-Magnitude Rows: Unscaled		Single-Magnitude Rows: Scaled	
ETA	275	155	(56%)	31	(11%)	121	(44%)
GIF-PIN	590	590	(100%)	264	(45%)	511	(87%)
RECIPE	82	53	(65%)	49	(60%)	52	(63%)
SCAGR7	89	63	(71%)	46	(52%)	50	(56%)
SCFXM1	289	129	(45%)	86	(30%)	111	(38%)
SCSD1	77	39	(51%)	0	(0%)	39	(51%)
SCTAP1	300	126	(42%)	120	(40%)	126	(42%)
SC205	203	147	(72%)	109	(54%)	120	(59%)
STAIR	356	165	(46%)	61	(17%)	159	(45%)
VTPBASE	166	63	(38%)	20	(12%)	51	(31%)

In his master's thesis, Kyu-Ho Ahn studied various network-extraction heuristics. He carried out computational studies to compare the following five algorithms:

(1) The method of Brown and Wright--This is a delection method: it starts with the set of all candidate rows, and deletes confilicting rows until a network is found.

(2) A revised version of the Brown-Wright method that omits certain updating operations at each step.

(3) *A method of* Gunawardane, Hoff and Schrage which is also a deletion method, extended to allow reflection of rows.

(4) A revised version of an extended Gunawardane-Hoff-Schrage method that omits certain updating operations at each step.

(5) A new insertion method, which starts with an empty network-row set and adds to it as the columns are scanned.

Each of these was implemented to incorporate a second phase, suggested by Brown and Wright, which attempts to simply add further candidate rows to the network set found by the first phase.

All of the above algorithms displayed comparable effectiveness in finding a large network subset:

		BW Method		GHS Method		
LP name	scaled?	Orig.	Rev.	Orig.	Rev.	Method of Ahn
ETA	yes	157	143	157	147	164
SCAGR7	no	83	81	83	81	83
SCAGR7	yes	83	81	82	80	83
SC205	no	78	110	78	78	78
SC205	yes	96	112	96	96	96
STAIR	no	59	58	59	57	59
STAIR	yes	155	154	155	153	155
VTPBASE	no	40	39	40	39	40
VTPBASE	yes	44	42	42	38	43

However, the algorithm varied greatly in the amount of computation required. The following timings (in central-processor seconds) were *made on a* Cyber 170/730:

		BW Method		GHS Method		
LP name	scaled?	Orig.	Rev.	Orig.	Rev.	Method of Ahn
ETA	yes	18.79	2.13	20.31	4.49	1.18
SCAGR7	no	.65	.35	1.02	.42	.19
SCAGR7	yes	1.34	.52	2.72	.62	.33
SC205	no	3.75	.74	8.34	.88	.51
SC205	yes	2.67	.71	6.88	.82	.49
STAIR	no	1.21	.88	1.51	.94	.93
STAIR	yes	1.58	1.24	2.41	1.19	1.07
VTPBASE	no	.78	.43	1.36	.58	.37
VTPBASE	yes	2.37	.61	3.24	.67	.40

Here Ahn's insertion method seems generally preferable. (The numbers of network rows above differ from those given in the previous tables, due to differences in scalings and in numbers of inessential rows deleted.)

An "exchange" algorithm was designed to follow any of the above five. It attempts to identify a row in the network subset that, when deleted, allows two or more ±1 rows to be added. This algorithm made substantial additions to the row subsets for SC205, SCAGR7 and ETA in some cases. The size of the augmented row subset, when the exchange algorithm ran after each of the five above, was as follows:

		BW Method		GHS Method		
LP name	scaled?	Orig.	Rev.	Orig.	Rev.	Method of Ahn
ETA	yes	160	157	160	157	164
SCAGR7	no	83	83	83	83	83
SCAGR7	yes	88	88	88	88	88
SC205	no	110	110	110	110	110
SC205	yes	120	113	120	120	120
STAIR	no	59	58	59	58	59
STAIR	yes	155	154	155	154	155
VTPBASE	no	41	41	41	41	41
VTPBASE	yes	44	44	44	44	44

Thus the effectiveness of the five algorithms was nearly identical when followed by the exchange algorithm. However, the exchange algorithm added substantially to the computation times:

		BW Method		GHS Method		
LP name	scaled?	Orig.	Rev.	Orig.	Rev.	Method of Ahn
ETA	yes	24.66	10.02	23.05	7.37	2.70
SCAGR7	no	1.02	.66	.65	.59	.19
SCAGR7	yes	3.82	1.80	2.30	1.53	1.50
SC205	no	11.75	1.93	7.17	5.91	6.19
SC205	yes	9.62	3.62	5.32	4.84	4.59
STAIR	no	1.85	1.29	1.58	1.27	1.27
STAIR	yes	3.12	1.88	2.27	2.01	1.74
VTPBASE	no	1.97	.22	1.44	1.10	.96
VTPBASE	yes	3.69	1.47	3.38	1.57	1.28

Here it is not so clear which algorithm is best. Also, further experimentation will be required to determine whether the expense of the exchange algorithm is worth the increase in network rows.

Finally, we applied the algorithm of Bixby and Cunningham to test whether certain subsets of rows were hidden networks. In most cases, a hidden-network subset was constructed by adding rows to a known embedded-network subset. The sizes of the largest embedded and hidden networks discovered were as follows:

LP name	Rows	Embedded-Network Rows	Hidden-Network Rows
ETA	275	121 (44%)	123 (45%)
GIF-PIN	590	511 (87%)	590 (100%)
RECIPE	82	52 (63%)	53 (65%)
SCAGR7	89	50 (56%)	61 (69%)
SCFXM1	289	111 (38%)	125 (43%)
SCSD1	77	39 (51%)	39 (51%)
SCTAP1	300	126 (42%)	126 (42%)
SC205	203	120 (59%)	147 (72%)
STAIR	356	159 (45%)	165 (46%)
VTPBASE	166	51 (31%)	52 (31%)

Thus notablty larger hidden networks were found in SCARG7, SCFXM1 and SC205, and GIF-PIN was found to be a **pure network**!

3. Mr. Ahn is currently working on developing methods for exploiting embedded structures computationally, once they have been found. Since there exist very efficient simplex methods for full-network linear programs, it is reasonable to seek efficient simplex methods for linear programs that have large partial networks. Previous studies of partial-network simplex methods (also often called embedded-network simplex methods) have taken either a decomposition or a working basis approach. Ahn is taking a more straigtforward approach, in which the routines of the simplex algorithm are adapted to take advantage of the partial network.

He will be particularly interested in applying partial-network simplex methods to embedded- or hidden-network LPs, and will initially concentrate on row-wise partial-networks LPs. These offer several advantages. In such LPs, the size of the row-wise partial network is the same in every basis. Thus basis updates are simplified, and the partial-network structure is equally advantageous at every iteration. In the hidden-network case, moreover, elementary row operations can transform some rows to a partial network without disturbing the non-network part of the matrix.

4. The <u>graph-realization</u> problem may be described as follows: Given a finite set E and a collection P of noncomparable subsets of E, when is there a tree T with edge-set E such that P is a collection of edge sets of paths in T? The applications of this problem to finding hidden-network structure in linear programs, an idea originally due to Iri, is by now well known, and forms the basis for much of the work described above. In his thesis, Don Wagner presents an algorithm for this problem which is, in essence, as nearly linear as possible (linear in the size of E), where a linear time bound is an obvious lower bound on problem complexity. His algorithm is the fastest one known.

Paper [3] gives a survey of current work on the graph-realization problem and its applications to structure problems in linear programming. It also contains a new algorithm for recognizing network structure in "implicit matrices", matrices given by an oracle that determines column independence. This problem was originally solved by P.D. Seymour, and is significantly more subtle than the explicit matrix version. In addition, a short proof a strengthened version of Seymour's result is given. This stronger result is due to Truemper.

5. Call a matrix <u>totally triangular</u> if every maximal linearly independent set of columns, that is, every <u>basis</u>, can be permuted to an upper triangular matrix. Collette Coullard is working on the problem of recognizing such matrices. Several partial results have been obtained. For example, we have shown that if the given matrix contains a basis that can be permuted to an identity, then it must necessarily happen that the entire matrix A can be scaled to a totally unimodular matrix; moreover, there is an efficient algorithm to find this scaling.

6. Let G be an undirected graph, and let e be a distinguished edge of G. Let **C** be the family of circuits of G that contain e, and let **P** = {C - e: C ∈ **C**}. Let A be a (0,1)-matrix the rows of which are the incidence vectors of the members of P. We call such a matrix a <u>path matrix</u>. Path matrices arise as constraint matrices in the so-called path-arc formulation of the classical maximum-flow problem, and, computationally, are relevant to the multicommodity flow problem. In [5] an algorithm is given for recognizing path matrices in polynomial time, and an algorithm is also given for recognizing the more general class of "totally Mengarian" matrices, for which the max-flow min-cut equality also holds. The problem of recognizing "dipath matrices" of directed graphs is left open, although significant progress has been made by E. Balas and a student, at Carnagie-Mellon.

C. Perfect Graphs

The vertex packing problem is the graph theoretic form of the well-known integer programming set packing problem. A <u>perfect graph</u> is graph for which we have an efficient method to verify optimality in the associated vertex packing problem. In [2] a method is described for composing two given perfect graphs to form a larger perfect graph.

D. Packing and Covering

Polynomial-time algorithms are presented in [6] for solving combinatorial packing and covering problems defined by the integral feasible flows in an integral supply-demand network.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
-

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFOSR-TR- 84-0046

6a. NAME OF PERFORMING ORGANIZATION
Northwestern University

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Air Force Office of Scientific Research

6c. ADDRESS *(City, State and ZIP Code)*
Department of Industrial Engineering
Evanston IL 60201

7b. ADDRESS *(City, State and ZIP Code)*
Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
AFOSR

8b. OFFICE SYMBOL *(If applicable)*
NM

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
AFOSR-82-0004

8c. ADDRESS *(City, State and ZIP Code)*
Bolling AFB DC 20332

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2304	A6	

11. TITLE *(Include Security Classification)* "HIDDEN AND EMBEDDED STRUCTURE IN LINEAR PROGRAMS"

12. PERSONAL AUTHOR(S)
Robert E. Bixby

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM 1/10/81 TO 30/9/83

14. DATE OF REPORT *(Yr., Mo., Day)*
SEP 30, 1983

15. PAGE COUNT
8

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The report is a summary of work completed on the grant, and begins with a listing of papers written followed by further descriptions of work completed as well as work in progress. During this period, six technical papers were completed and submitted to refereed journals. Titles were — "A simple theorem on 3-connectivity," "A composition for perfect graphs," "The partial order of a polymatroid extreme point," "Algorithms for two versions of graph realization and an application to linear programming," "A note on recognizing path matrices," and "Packing and covering by integral feasible flows in integral supply-demand networks."

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☒ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Dr. Joseph Bram

22b. TELEPHONE NUMBER *(Include Area Code)*
(202) 767- 4939

22c. OFFICE SYMBOL
[illegible]

DD FORM 1473, 83 APR

EDITION OF 1 JAN 73 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

END

DATE
FILMED

3 84

DTIC