Conf-9306195--15

# Applications of an Algebraic Monge Property
(extended abstract)

Wolfgang W. Bein * Peter Brucker † James K. Park ‡

May 1, 1993


## Abstract

When restricted to cost arrays possessing the sum Monge property, many combinatorial optimization problems with sum objective functions become significantly easier to solve. (An array $A = \{a[i,j]\}$ possesses the sum Monge property if for all $i < k$ and $j < \ell$, $a[i,j] + a[k,\ell] \leq a[i,\ell] + a[k,j]$.) Examples include the usual sum-objective-function versions of the assignment problem, the transportation problem, the traveling-salesman problem, and several shortest-path problems. Furthermore, the more general algebraic assignment and transportation problems, which are formulated in terms of an ordered commutative semigroup $(H, *, \preceq)$, are similarly easier to solve given cost arrays possessing the corresponding algebraic Monge property, which requires that for all $i < k$ and $j < \ell$, $a[i,j] * a[k,\ell] \preceq a[i,\ell] * a[k,j]$.

In this paper, we show that Monge-array results for two sum-of-edge-costs shortest-path problems can likewise be extended to a general algebraic setting, provided the problems' ordered commutative semigroup $(H, *, \preceq)$ satisfies one additional restriction. Specifically, we require that for all $a, b, c \in H$, $a \prec b$ implies $c * a \prec c * b$. In addition to this general result, we also show how our algorithms can be modified to solve certain bottleneck shortest-path problems, even though the ordered commutative semigroup $(\Re, \max, \leq)$ naturally associated with bottleneck problems does not satisfy our additional restriction. The bottleneck shortest-path problems we can solve are those with cost arrays possessing what we call the strict bottleneck Monge property, which requires that for all $i < k$ and $j < \ell$, either $\max\{a[i,j], a[k,\ell]\} < \max\{a[i,\ell], a[k,j]\}$ or both $\max\{a[i,j], a[k,\ell]\} = \max\{a[i,\ell], a[k,j]\}$ and $\min\{a[i,j], a[k,\ell]\} \leq \min\{a[i,\ell], a[k,j]\}$. We also provide improved algorithms for several other bottleneck combinatorial optimization problems whose cost arrays possess the strict bottleneck Monge property. Finally, we show how our bottleneck shortest-path techniques can be used to obtain fast algorithms for (1) a variant of Hirschberg and Larmore's optimal paragraph formation problem, (2) a processor-allocation problem first formulated by Bokhari, and (3) a special case of the bottleneck traveling-salesman problem.


Many combinatorial optimization problems with sum objectives have efficient algorithms for algebraic objective functions, see e.g. Burkard and Zimmermann [10], Burkard [7], Burkard [8], Seiffart [18]. In the sum case often substantial efficiency gain is possible when the underlying costs have the Monge property. In this paper we will derive such results for two path problems with algebraic cost arrays.

Consider here the complete directed acyclic graph $G = (V, E)$, i.e. G has vertices $V = \{1, \ldots, n\}$ and edges $(i,j) \in E$ iff $i < j$. Associated with the edges are costs $a[i,j]$, which are drawn from an ordered commutative semigroup $(H, *, \preceq)$. We require that the internal composition $*$ be strictly

*American Airlines Decision Technologies, MD 4462 HDQ, P. O. Box 619616, Dallas/Fort Worth Airport, TX 75261-9616, U.S.A. Electronic mail: wolf@aadt.com.

†Fachbereich Mathematik/Informatik, Universität Osnabrück, Postfach 44 69, D-4500 Osnabrück, Germany. Electronic mail: pbrucker@dosuni1.rz.uni-osnabrueck.de.

‡Algorithms and Discrete Mathematics Department (Org. 1423), Sandia National Laboratories, P. O. Box 5800, Albuquerque, NM 87185-5800, U.S.A. Electronic mail: jkpark@cs.sandia.gov. This author's work was supported by the U.S. Department of Energy under Contract DE-AC04-76DP00789.

compatible with the order relation $\leq$, i.e., for all $a, b, c \in H$, $a \prec b$ implies $c * a \prec c * b$ (this additional property is essential, as we will see later.) The algebraic shortest-path problem is the problem of finding the shortest path from vertex 1 to vertex $n$ whereas the k-edge algebraic shortest-path problem is the problem of finding such a path that has exactly $k$ edges. We also show how our results relate to bottleneck objective functions for these problems; see Gabow and Tarjan [13] for background concerning bottelneck shortest path problems.

Considering the ordered commutative subgroup $(\Re, \max, \leq)$ naturally associated with bottleneck combinatorial optimization problems we note that the composition max is compatible with the order relation $\leq$ but not strictly compatible with it. (For example, $5 < 7$ but $\max\{8,5\} \not< \max\{8,7\}$.)

For an example of an ordered commutative semigroup $(H, *, \preceq)$ whose internal composition $*$ is strictly compatible with its order relation $\preceq$, consider the set $T$ of ordered tuples $(r_1, r_2, \ldots, r_n)$ such that $n \geq 0$, $r_i \in \Re$ for $1 \leq i \leq n$, and $r_1 \leq r_2 \leq \cdots \leq r_n$. Furthermore, suppose we define $\oplus$ so that

$$(q_1, q_2, \ldots, q_m) \oplus (r_1, r_2, \ldots, r_n) \;=\; (s_1, s_2, \ldots, s_{m+n}) \;,$$

where $s_1, s_2, \ldots, s_{m+n}$ is the sorted sequence obtained by merging the sequences $(q_1, q_2, \ldots, q_m)$ and $(r_1, r_2, \ldots, r_n)$, and $\prec$ so that

$$(q_1, q_2, \ldots, q_m) \prec (r_1, r_2, \ldots, r_n)$$

if and only if there exists an $i$ in the range $1 \leq i \leq m$ such that $q_i < r_i$ and $q_j = r_j$ for $1 \leq j < i$ *or* $m < n$ and $q_j = r_j$ for $1 \leq j \leq m$. It is not hard to see that $(T, \oplus, \preceq)$ is an ordered commutative semigroup and $\oplus$ is strictly compatible with $\preceq$. As we will see later this semigroup can be used to model strict bottleneck Monge conditions.

We will now show that both the unrestricted and $k$-edge variants of the algebraic shortest-path problem for an ordered commutative semigroup $(H, *, \preceq)$ are significantly easier to solve given edge costs with the algebraic Monge property, provided the internal composition $*$ is strictly compatible. (An array $A = \{a[i,j]\}$ possesses the algebraic Monge property if for all $i < k$ and $j < \ell$, $a[i,j] * a[k,\ell] \preceq a[i,\ell] * a[k,j]$.) Strict compatibility is necessary to insure that every array possessing the algebraic Monge property also exhibits total monotonicity, [1] the crucial property exploited by our algorithms. The following lemma makes this last claim precise.

**Lemma 1** Let $(H, *, \preceq)$ denote an ordered commutative semigroup whose internal composition $*$ is *strictly* compatible with its order relation $\preceq$, and let $A = \{a[i,j]\}$ denote an array whose entries are drawn from $(H, *, \preceq)$. If $A$ possesses the algebraic Monge property, then $A$ is totally monotone. ∎

Note that if the semigroup's composition $*$ is compatible with its order relation $\preceq$ but not strictly compatible with it, then an array whose entries are drawn from the semigroup may possess the algebraic Monge property without being totally monotone. For example, consider again the ordered commutative subgroup associated with bottleneck combinatorial optimization problems. The array

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

satisfies the inequality $\max\{a[i,j], a[k,\ell]\} \leq \max\{a[i,\ell], a[k,j]\}$ for all $i < k$ and $j < \ell$, but it is not totally monotone.

[1] An $m \times n$ array $A = \{a[i,j]\}$ is called totally monotone if for all $i$, $j$, $k$, and $\ell$ satisfying $1 \leq i < k \leq m$ and $1 \leq j < \ell \leq n$, either $a[i,j] \leq a[i,\ell]$ or $a[k,j] > a[k,\ell]$.

The total monotonicity of arrays possessing the algebraic Monge property allows us to locate these arrays' smallest entries using the array-searching algorithms of Aggarwal et al. [3] and Larmore and Schieber [15]. However, before we can obtain the desired shortest-path algorithms, we need one more lemma. (Note that this lemma does not require the strict-compatibility assumption.)

**Lemma 2** Let $(H, *, \preceq)$ denote an ordered commutative semigroup whose internal composition $*$ is compatible with its order relation $\preceq$, and let $A = \{a[i,j]\}$ denote an array whose entries are drawn from $(H, *, \leq)$. Furthermore, let $B = \{b[i]\}$ denote any vector, and let $C = \{c[i,j]\}$ denote the array given by $c[i,j] = b[i] * a[i,j]$. If $A$ possesses the algebraic Monge property, then so does $C$. ■

**Theorem 3** Let $(H, *, \preceq)$ denote an ordered commutative semigroup whose internal composition $*$ is strictly compatible with its order relation $\preceq$, and let $G$ denote a complete directed acyclic graph on vertices $1, \ldots, n$ whose edge costs are drawn from $H$. If $G$'s edge costs possess the algebraic Monge property, then the algebraic $k$-edge shortest-path problem for $G$ can be solved in $O((t_a + t_c)kn)$ time, where $t_a$ is the worst-case time required for computing $d_{\ell-1}[i] * c[i,j]$ and $t_c$ is the worst-case time required for comparing two entries of $A_\ell$. ■

**Theorem 4** Let $(H, *, \preceq)$ denote an ordered commutative semigroup whose internal composition $*$ is strictly compatible with its order relation $\leq$, and let $G$ denote a complete directed acyclic graph on vertices $1, \ldots, n$ whose edge costs are drawn from $H$. If $G$'s edge costs possess the algebraic Monge property, then the algebraic unrestricted shortest-path problem for $G$ can be solved in $O((t_a + t_c)n)$ time, where $t_a$ is the worst-case time required for computing $d[i] * c[i,j]$ and $t_c$ is the worst-case time required for comparing two entries of $A$. ■

As mentioned earlier, the ordered commutative subgroup $(\Re, \max, \leq)$ naturally associated with bottleneck combinatorial optimization problems has a composition that is not strictly compatible with its order relation. Nevertheless, we will show how our two algebraic shortest-path algorithms can be modified to handle bottleneck shortest-path problems. Furthermore, we will present a second algorithm for the bottleneck $k$-edge shortest-path problem that in some sense generalizes Aggarwal, Schieber, and Tokuyama's [5] algorithm. This algorithm is based on a $O(n)$-time query subroutine for determining whether the graph contains a $k$-edge 1-to-$n$ using only edges whose costs are less than or equal to some threshold $T$. To obtain our results we assume that the cost array possesses what we call the strict bottleneck Monge property, which requires that for all $i < k$ and $j < \ell$, either $\max\{a[i,j], a[k,\ell]\} < \max\{a[i,\ell], a[k,j]\}$ or both $\max\{a[i,j], a[k,\ell]\} = \max\{a[i,\ell], a[k,j]\}$ and $\min\{a[i,j], a[k,\ell]\} \leq \min\{a[i,\ell], a[k,j]\}$.

**Theorem 5** The bottleneck unrestricted shortest-path problem for an $n$-vertex graph whose edge costs possess the strict bottleneck Monge property can be solved in $O(n)$ time. ■

**Theorem 6** The bottleneck $k$-edge shortest-path problem for an $n$-vertex graph whose edge costs possess the strict bottleneck Monge property can be solved in $O(kn)$ time. ■

**Theorem 7** The bottleneck $k$-edge shortest-path problem for an $n$-vertex graph whose edge costs possess the strict bottleneck Monge property can be solved in $O(n^{3/2} \lg^{5/2} n)$ time (or in $O(n \lg^2 n)$ time if the problem's cost array is also bitonic). [2] ■

Using a similar query technique, we can also obtain the following result for transportation problems with more total supply than demand.

[2] An $n$-entry vector $B = \{b[i]\}$ is called bitonic if there exists an $i$ satisfying $1 \leq i \leq n$ such that $b[1] \geq \cdots \geq b[i-1] \geq b[i] \leq b[i+1] \leq \cdots \leq b[n]$. We call a 2-dimensional array bitonic if its rows or its columns are bitonic.

**Theorem 8** The bottleneck (unbalanced) transportation problem for an $m \times n$ bipartite graph ($m \leq n$) whose edge costs possess the bottleneck Monge property can be solved in $O((m\sqrt{n \lg m} + n)\lg^2 n)$ time (or in $O(m \lg^2 n)$ time if the problem's cost array is also bitonic). ∎

We conclude with three applications of our bottleneck-shortest-path techniques. The first is a variant of Hirschberg and Larmore's optimal-paragraph-formation problem [14]. They considered the problem of breaking a sequence of words $w_1, \ldots, w_n$ into lines (i.e., subsequences) in order to form a paragraph. Roughly speaking, they defined the cost $f(i,j)$ of a line consisting of words $w_i$ through $w_{j-1}$ to be the square of the difference between this line's length and the ideal line length, and their objective was to construct a paragraph minimizing the sum of the paragraph's line costs. This problem is easily transformed into an instance of the sum unrestricted shortest-path problem with edge costs possessing the sum Monge property, and thus it can be solved in $O(n)$ time. (Credit for the linear-time algorithm belongs to Wilber [19].) If we instead seek to minimize the maximum line cost, we obtain an instance of the bottleneck unrestricted shortest-path problem with edge costs possessing the strict bottleneck Monge property; thus, by Theorem 5, this natural variant of Hirschberg and Larmore's problem can also be solved in $O(n)$ time. Furthermore, this last result can be generalized by observing that the shortest-path problem's edge costs possess the strict bottleneck Monge property for any line cost function $f(i,j)$ that is strictly bitonic (i.e., strictly decreasing then strictly increasing) in both $i$ and $j$. We can also handle a variant of the bottleneck problem that allows hyphenation (with a penalty, of course), even though the associated line cost function may not be bitonic.

Our second application involves a processor-allocation problem first formulated by Bokhari [6]. We are given a chain of $m$ modules (corresponding to a parallel or pipelined computation), and we want to map these modules onto a chain of $n$ processors, where $m > n$. We assume that processor $k$ can execute modules $i$ through $j-1$ in $t_k[i,j] = c_k[i] + c_k[j] + \sum_{\ell=i}^{j} w_k[\ell]$ time, where $w_k[i]$ is the time to execute the $i$th module on processor $k$ and $c_k[i]$ is the the communication cost associated with mapping modules $i-1$ and $i$ to different processors, and we seek to minimize the maximum of the processor execution times. Bokhari gave an $O(m^3 n)$-time algorithm for this problem, and Nicol and O'Hallaron [16] improved its running time to $O(m^2 n)$. Nicol and O'Hallaron also gave an $O(mn \lg m)$-time algorithm for a special case where (1) the processors and their communication links are homogeneous (i.e., $w_k[i] = w[i]$ and $c_k[i] = c[i]$ for all $i$) and (2) there exist constants $\mathcal{W}$ and $\mathcal{C}$ such that $w[i] > \mathcal{W} > 0$ and $0 \leq c[i] < \mathcal{C}$ for all $i$. We will show that Bokhari's problem can be solved in $O(mn + m \lg m)$ time (without any assumptions) using a slight generalization of Theorem 6. The key to obtaining this result is rearranging the rows and columns of $T_k = \{t_k[i,j]\}$ to obtain an array possessing the strict bottleneck Monge property. (A similar technique is used by Aggrawal and Park [4].)

For our final application, we consider a special case of the bottleneck traveling-salesman problem. Given a complete directed graph $G$ on vertices $\{1, \ldots, n\}$ and a cost array $C = \{c[i,j]\}$ assigning cost $c[i,j]$ to the edge $(i,j)$, we seek a tour of $G$ that visits every vertex of $G$ exactly once and minimizes the maximum of the tour's edges' costs. In [9], Burkard and Sandholzer identified several families of cost arrays corresponding to graphs containing at least one bottleneck-optimal tour that is pyramidal. (A tour $T$ is called pyramidal if (1) the vertices on the path $T$ follows from vertex $n$ to vertex 1 have monotonically decreasing labels, and (2) the vertices on the path $T$ follows from vertex 1 to vertex $n$ have monotonically increasing labels.) Thus, since there is a simple $O(n^2)$-time dynamic-programming algorithm for computing a pyramidal tour whose maximum edge cost is minimum among all pyramidal tours, the bottleneck traveling-salesman problem for any graph whose cost array is a member of one of Burkard and Sandholzer's families can be solved in $O(n^2)$ time. Using a generalization of Theorem 5, we will show that if a graph's edge cost array possesses the strict bottleneck Monge property, then it is possible to find the graph's best pyramidal tour in $O(n)$ time. Thus, since one of Burkard and Sandholzer's families consists of arrays possessing the bottleneck Monge property (see Corollary 4.5 of their paper), we can solve the bottleneck traveling-salesman problem for any graph with edge costs possessing the strict bottleneck Monge property in $O(n)$ time. (This result is analogous to the result proved by Park [17] for the usual sum traveling-salesman problem.)

## References


[1] I. Adler, A. J. Hoffman, and R. Shamir. Monge and feasibility sequences in general flow problems. *Discrete Applied Mathematics*, 1993. To appear.

[2] P. K. Agarwal and S. Sen, June 1992. Personal communication.

[3] A. Aggarwal, M. M. Klawe, S. Moran, P. W. Shor, and R. Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2(2):195–208, 1987.

[4] A. Aggarwal and J. K. Park. Improved algorithms for economic lot-size problems. *Operations Research*, 1993. To appear.

[5] A. Aggarwal, B. Schieber, and T. Tokuyama. Finding a minimum weight $K$-link path in graphs with Monge property and applications. In *Proceedings of the 9th Annual ACM Symposium on Computational Geometry*, 1993. To appear.

[6] S. H. Bokhari. Partitioning problems in parallel, pipelined, and distributed computing. *IEEE Transactions on Computers*, 37(1):48–57, 1988.

[7] R. E. Burkard. Remarks on some scheduling problems with algebraic objective functions. *Operations Research Verfahren*, 32:63–77, 1979.

[8] R. E. Burkard. A note on greedily solvable time transportation problems. Report 243, Institut für Mathematik, Technische Universität Graz, Graz, Austria, November 1992.

[9] R. E. Burkard and W. Sandholzer. Efficiently solvable special cases of bottleneck travelling salesman problems. *Discrete Applied Mathematics*, 32(1):61–76, 1991.

[10] R. E. Burkard and U. Zimmermann. Combinatorial optimization in linearly ordered semimodules: A survey. In B. Korte, editor, *Modern Applied Mathematics: Optimization and Operations Research*, pages 391–436. North-Holland Publishing Company, Amsterdam, 1982.

[11] G. N. Frederickson. Optimal algorithms for tree partitioning. In *Proceedings of the 2nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 168–177, 1991.

[12] G. N. Frederickson and D. B. Johnson. The complexity of selection and ranking in X + Y and matrices with sorted columns. *Journal of Computer and System Sciences*, 24(4):197–208, 1982.

[13] H. N. Gabow and R. E. Tarjan. Algorithms for two bottleneck optimization problems. *Journal of Algorithms*, 9(3):411–417, 1988.

[14] D. S. Hirschberg and L. L. Larmore. The least weight subsequence problem. *SIAM Journal on Computing*. 16(4):628–638, 1987.

[15] L. L. Larmore and B. Schieber. On-line dynamic programming with applications to the prediction of RNA secondary structure. *Journal of Algorithms*, 12(3):490–515, 1991.

[16] D. M. Nicol and D. R. O'Hallaron. Improved algorithms for mapping pipelined and parallel computations. *IEEE Transactions on Computers*, 40(3):295–306, 1991.

[17] J. K. Park. A special case of the $n$-vertex traveling-salesman problem that can be solved in $O(n)$ time. *Information Processing Letters*, 40(5):247–254, 1991.

[18] E. Seiffart. Algebraic transportation and assignment problems with "Monge-property" and "quasi-convexity". *Discrete Applied Mathematics*, 1993. To appear.

[19] R. Wilber. The concave least-weight subsequence problem revisited. *Journal of Algorithms*. 9(3):418–425, 1988.

# END

# DATE FILMED

9/17/93