Classic Optimization Techniques Applied to Stormwater and Nonpoint Source Pollution Management at the Watershed Scale

James F. Limbrunner, M.ASCE[1]; Richard M. Vogel, M.ASCE[2]; Steven C. Chapra, F.ASCE[3]; and Paul H. Kirshen, M.ASCE[4]

Abstract: Linear and dynamic programming formulations are introduced for optimizing the placement of distributed best management practices (BMPs) at the watershed scale. The results of linear programming optimization of infiltration-based stormwater management BMPs are compared with the results of genetic algorithm (GA)optimization using a nonlinear distributed model. Additionally, linear and dynamic programming optimization of sediment-trapping BMPs are compared with GA optimization using a nonlinear distributed model. The results indicate that the solution to stormwater peak-flow reduction is influenced primarily by distributed-flow arrival time, and a linear programming analog to a nonlinear optimization model can efficiently reproduce much of the same solution structure. Linear and dynamic programming solutions to the storm sediment-management problem indicate natural sediment trapping is an important consideration, and a solution to the sediment-management-optimization problem can be efficiently found using a dynamic programming formulation. **DOI: 10.1061/(ASCE)WR.1943-5452.0000361.** *© 2013 American Society of Civil Engineers.*

CE Database subject headings: Watersheds; Nonpoint pollution; Best management practice; Stormwater management; Sediment loads; Optimization.

Author keywords: Watershed management; Nonpoint source pollution; Best management practices; Stormwater; Sediment load; Linear programming; Dynamic programming.

Introduction

Simulation models are widely used for studying watershed best management practice (BMP) system behavior, and it is likely that their use will continue in developing total maximum daily loads (TMDLs) and in designing solutions to nonpoint source pollution problems (National Research Council 2001). Stormwater and nonpoint load generation mechanisms are complex, as are the models used to simulate them. Using complex simulation models for management alternative scenario testing is a popular approach to designing stormwater and nonpoint source pollution management systems, whereby various management options are tested in a simulation model, and the resulting stormwater and nonpoint source loads are compared for selection of a management alternative. Where design choices may be limited by other factors, such as available locations for construction, testing a few practical alternatives using a simulation model is often a satisfactory approach. But there may be other cases where flexibility in construction sites, technology selection, and budget make many combinations of alternatives possible. For these less-constrained design situations, testing a few scenarios may not be adequate to find a near-optimal solution. Contemporary optimization techniques, such as GAs and other evolutionary algorithms, have been employed to perform more comprehensive searches of large decision spaces and have become popular for analyzing nonpoint source pollution management designs. While contemporary optimization techniques expand the horizons for evaluation of nonpoint source pollution management alternatives, the computational time needed by the algorithms to perform many runs of a complex simulation model can be burdensome.

Long waits for results may limit the practicality of using evolutionary algorithms in typical forums for collaborative decision making, such as stakeholder workshops and brainstorming sessions, which lie at the heart of navigating the often contentious process of making decisions regarding natural resources. Since convening large groups of stakeholders is expensive, real-time responses to questions and suggestions are desirable to make the most of workshops and to move a decision process forward. The many runs of computationally intensive simulation models, which are required for genetic algorithm optimization, when combined with the need for real-time stakeholder involvement, present a conundrum regarding the selection of tools for natural resource decision-making processes. Simulation models are needed to adequately describe many of a watershed's physical processes, and at the same time, fast analyses and optimizations are needed to facilitate collaborative processes for negotiating complex decisions among groups of stakeholders. An approach that combines rapid screening of ideas during a stakeholder workshop, followed

[1]Senior Water Systems Analyst, HydroLogics, Venture Development Center, Univ. of Massachusetts Boston, Boston, MA 02125 (corresponding author). E-mail: jlimbrunner@hydrologics.net

[2]Professor, Dept. of Civil and Environmental Engineering, Tufts Univ., Medford, MA 02155.

[3]Professor and Berger Chair, Dept. of Civil and Environmental Engineering, Tufts Univ., Medford, MA 02155.

[4]Research Professor, Environmental Research Group, Civil Engineering Dept. and Institute for the Study of Earth, Oceans, and Space, Univ. of New Hampshire, Durham, NH 03824.

Note. This manuscript was submitted on October 17, 2011; approved on February 14, 2013; published online on August 15, 2013. Discussion period open until February 1, 2014; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Water Resources Planning and Management***, Vol. 139, No. 5, September 1, 2013. © ASCE, ISSN 0733-9496/2013/5-486-491/$25.00.

by the refinement of results using a detailed simulation model, may allow for the right balance between productivity and accuracy.

Classic optimization techniques, such as linear programming and dynamic programming, have not been widely applied to the contemporary problems of nonpoint source pollution management optimization, though they have potential for overcoming the limitation, at the screening stage, that is imposed by long computation times needed by GA optimization. The use of classic optimization techniques has potential for improving the productivity of stakeholder collaboration.

There is a small body of literature regarding the application of classic optimization techniques to stormwater and nonpoint source pollution management. Mays and Bedient (1982) developed a dynamic program for locating and sizing ponds for dendritic detention systems with drainage channels connecting detention basins. Jenq et al. (1983) used a linear program to analyze nonpoint source pollution reduction following the type of analysis performed by ReVelle et al. (1968). In their analysis, Jenq et al. (1983) lumped nonpoint source pollutant generating areas and treated the discharge from these areas as point sources of pollution to a stream. Schleich and White (1997) used linear programming and a lumped watershed modeling approach to find the least-cost management strategy to meet phosphorus and total suspended solids reduction requirements for a watershed containing both point sources and nonpoint sources. Sample et al. (2001) investigated the optimal mix of BMP controls at a site scale using linear programming to calculate minimum cost land-use options that met development goals while retaining predevelopment rainfall initial abstraction.

The goal of this study is to further explore the application of classic optimization techniques within the contemporary context of distributed stormwater and nonpoint source pollution management by comparing the results of distributed simulation models and a GA optimization approach with the results obtained by classic optimization techniques.

Linear Programming Optimization of Infiltration-Based BMP Placement

Perez-Pedini et al. (2005) developed a distributed watershed model using the Soil Conservation Service (SCS) curve number method (USDA 1986) that was similar to the work of Moglen (2000), who showed that a modified SCS curve number method could be distributed on a watershed network, accounting for the transport of water from upslope cells to downslope cells. Perez-Pedini et al. (2005) modeled a small urbanized watershed with a distributed model comprised of 1.44-ha connected hydrologic response units. Overland flow routing was simulated by passing runoff from each cell to the adjacent cell with the lowest elevation during each time step. An assumption was made that sewers do not alter drainage direction and arrival time. Flow routing in the channel network was modeled similarly to surface routing, but a shorter travel time was assumed, and water was passed a distance of two stream cells during each time step. Because of this simplified flow-routing scheme, arrival time of overland flow at the basin outlet is fully defined by network connectivity. The lag between flow generation at any cell in the watershed and arrival at the outlet is

$$\phi_i = (n_i - 1) + \mathrm{floor}\left(\frac{m_i}{2}\right) \tag{1}$$

where (n_i) = number of land surface cells between cell i and a stream cell; and m_i = number of stream cells between the basin outlet and the point where stormwater from cell i enters the stream network. A surface flow hydrograph $Q(t)$ at the watershed outlet can be created by summing all contributing flows at each time step:

$$Q(t) = \sum_i q(i, t - \phi_i) \tag{2}$$

where $q(i, t)$ = flow generated in cell i at time t; and ϕ_i = time lag between flow generation at cell i and arrival at the outlet.

In the GA optimization from Perez-Pedini et al. (2005), the goal was to find optimal arrangements of infiltration-based BMPs. A BMP, if implemented, improved the infiltration capacity of a model cell and was represented as a reduction of the SCS curve number.

The goal of the analysis described here was to explore whether a linear programming formulation could be used to reproduce the optimization results found by Perez-Pedini et al. (2005). In the present analysis, the question is not whether a linear approximation can be found for the rainfall-runoff simulation model but rather whether a linear program can be used to reproduce the optimal decision regarding where BMPs should be placed for maximum effectiveness in reducing the stormwater-runoff peak. The analysis begins with the extraction of flow generated within each of the cells in the nonlinear simulation model developed by Perez-Pedini et al. (2005) for a storm occurring on October 15, 2003. Surface flow $q(i, t)$ is assumed to be the sum of runoff from precipitation and saturation excess $(SE_{i,t})$, once the initial abstraction at each cell has been satisfied:

$$q(i, t) = \frac{P_t^2}{P_t + S_{\max,i}} + SE_{i,t-1} \tag{3}$$

where (P_t) = precipitation; and $(S_{\max,i})$ = soil moisture storage capacity.

In the linear programming optimization, the placement of a BMP in a cell is represented by assuming a 10% reduction of flow generated within that cell. This stylistic representation of a BMP is similar to the approach taken by Perez-Pedini et al. (2005). The linear model is

$$\text{Minimize } Q_{\max} \text{ subject to} \tag{4}$$

$$Q(t) = \sum_i [q(i, t - \phi_i) - 0.1 \cdot \mathrm{BMP}_i \cdot q(i, t - \phi_i)] \quad \forall\ t \tag{5}$$

$$Q_{\max} \geq Q(t) \quad \forall\ t \tag{6}$$

$$0 \leq \mathrm{BMP}_i \leq 1 \quad \forall\ i \tag{7}$$

Perez-Pedini et al. (2005) allowed BMP placement in only 1,908 of the 4,533 cells in their study, based on the criteria of high curve number and close proximity to the stream. The constraint in Eq. (7) was applied to the same 1,908 cells allowed by Perez-Pedini et al. (2005), but at other locations, BMPs were not allowed. The overall budget constraint requires that the total number of BMPs applied to the watershed does not exceed a selected budget, expressed as the maximum number of BMPs, (B):

$$\sum_i \mathrm{BMP}_i \leq B \tag{8}$$

A continuous formulation was chosen to represent the integer decision problem to allow for solution with a linear programming algorithm. Results for the decision variables were integers. This may suggest the formulation is unimodular; however, it remains

J. Water Resour. Plann. Manage. 2013.139:486-491.

to be proven. Figs. 1(a–d) compare the solutions generated by Perez-Pedini et al. (2005) using a GA with their nonlinear optimization model and those found here using linear programming for budgets of $B = 25$, $B = 100$, $B = 150$, and $B = 400$ BMPs. The bottom image in each of Figs. 1(a–d) was created from the results reported in Perez-Pedini et al. (2005), and the top image in each figure presents the solution found using linear programming. The results generated by Perez-Pedini et al. (2005) required between approximately 15,000 and 73,000 iterations of the simulation model. The linear programming solution shown in the top image in each of Figs. 1(a–d) was found nearly instantaneously using *GAMS*.

The results of the linear optimization appear to capture many of the same clusters found using the GA and the nonlinear simulation model. Major clusters of BMP implementation found by Perez-Pedini et al. (2005) are circled in Fig. 1(d). One general difference appears to be a tendency for the linear programming solution to contain cells farther from the stream than did the GA solution in the nonlinear optimization. This is likely due to the absence of a representation of downslope infiltration in the linear optimization model. In the nonlinear simulation, overland flow may be infiltrated in downslope cells, but in the linear optimization all flow is routed directly to the watershed outlet with the appropriate time lag.

The similarity of the results obtained with the linear program and GA optimization supports the hypothesis that managing distributed-flow arrival time is the principal factor in storm-peak management. The similarity of the results also suggests that a linear programming model could be used as a screening tool for later GA optimization. By reducing the search space for a GA solution, more rapid optimization could be achieved. This two-step approach would combine the advantages of both linear programming and GA optimization using a nonlinear simulation model. Rapid screening-level optimization can be achieved with the linear program; then solutions can be refined and more accurately quantified using a simulation model and a GA.

Linear Programming Optimization of Sediment-Trapping BMP Placement

Limbrunner et al. (2013) extended the work of Perez-Pedini et al. (2005) with a similar distributed hydrology model based on 1.44-ha connected hydrologic response units and a sediment-generation and transport model based on the work of Jain et al. (2005). The results they obtained using a GA are compared with sediment-trapping BMP placement optimizations developed here using linear programming and dynamic programming. The comparison between optimization approaches using mathematical programming and a GA for the sediment-trapping problem is analogous to the stormwater peak-flow reduction problem described earlier. As previously, the goal of the analysis is to explore whether efficient optimization techniques can reproduce an optimal decision regarding where to place BMPs for maximum effectiveness.

There is a compound effect on sediment reduction when multiple BMPs lie in sequence along a slope line. Because of this, sediment generated in one location may be attenuated multiple times as it travels downslope to a stream. This effect was approximated in a linear programming formulation using a Lagrangian reference frame. A generated load was followed from its cell of origin to its point of discharge into a stream cell. The routing scheme for sediment load was the same as the routing scheme for stormwater described previously, with sediment load traveling from each cell to the adjacent cell with the lowest elevation. The nonlinear sediment-generation and transport model described in Limbrunner et al. (2013) was first used to sum all modeled sediment generated in each cell (l_i) during a storm event that occurred on August 18, 1992. Like the stormwater problem, where a simulation model was used first to quantify runoff generated in each cell, here a simulation model was used first to quantify the sediment load generated in each cell. Total load L reaching the stream during the storm can be expressed as

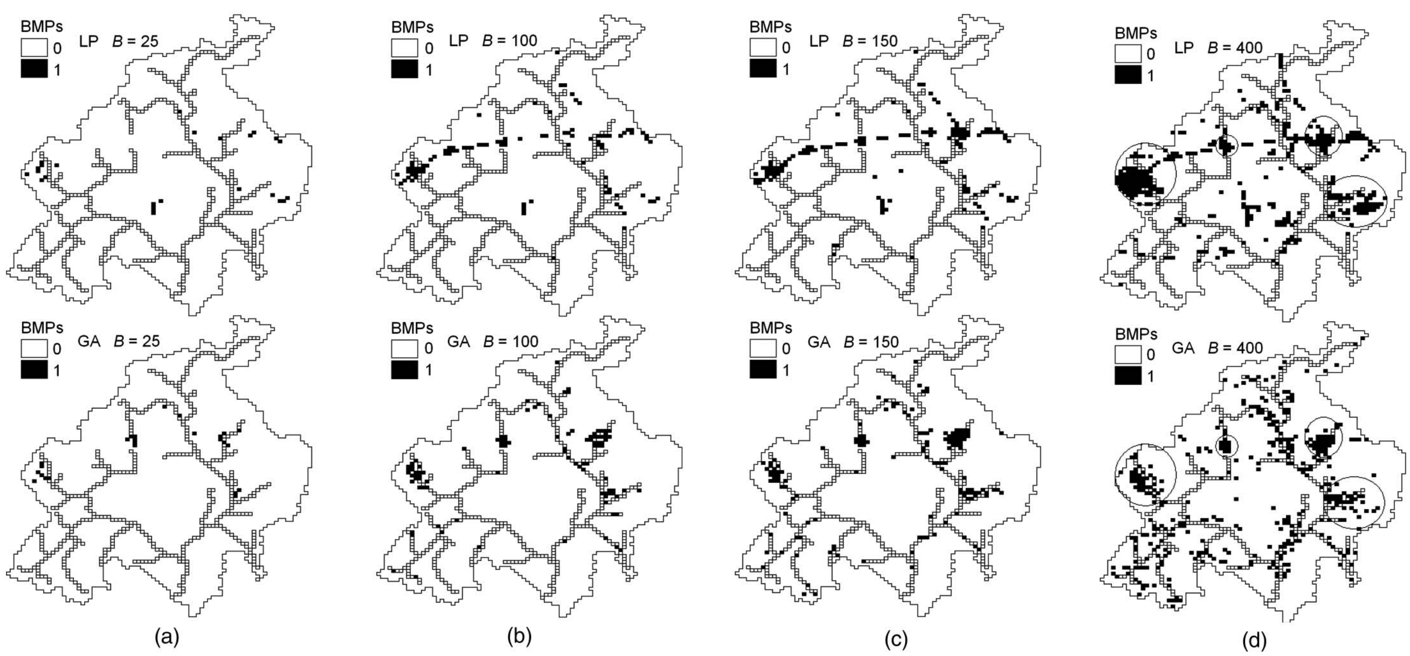

Fig. 1. Comparison of linear programming and genetic algorithm peak-flow reduction optimizations: (a) $B = 25$; (b) $B = 100$; (c) $B = 150$; (d) $B = 400$

J. Water Resour. Plann. Manage. 2013.139:486-491.

$$L = \sum_i l_i \left(1 - r \cdot \sum_{j(i)=0}^{n_i} \mathrm{BMP}_{i,j(i)}\right) \tag{9}$$

The reduction fraction (r) is preselected, and the placement of a BMP in any downslope cell $j(i)$ results in a reduction in sediment load (l_i) from cell i. This formulation results in a BMP attenuating a fixed fraction of sediment load (l_i) generated in cell i. The index $j(i)$ represents the jth downslope cell in the path that sediment travels from cell i to a stream cell. The index ranges from $j(i) = 0$, which is cell (i), to $j(i) = 1$, the cell immediately downslope from i, and so forth, to $j(i) = n_i$, which is the last cell in the slope line of i, and is directly connected to the stream network. Because of converging slope lines, cells may be common to more than one slope line, and a BMP may attenuate sediment load from many upslope cells. The linear program is formulated as

$$\text{Minimize } L \text{ subject to} \tag{10}$$

$$0 \leq \mathrm{BMP}_{i,0} \leq 1 \quad \forall\ i \tag{11}$$

$$\sum_{j(i)=0}^{n_i} \mathrm{BMP}_{i,j(i)} \leq \frac{1}{r} \quad \forall\ i \tag{12}$$

The constraint in Eq. (12) limits the number of BMPs along each (i) slope line and ensures the bracketed term on the right-hand side of Eq. (9) will remain positive. A value of $r = 0.5$ was chosen. The overall watershed BMP budget constraint is expressed as the maximum number of BMPs, (B):

$$\sum_i \mathrm{BMP}_{i,0} \leq B \tag{13}$$

Figs. 2(a–d) show a comparison of the results. The bottom image in each of Figs. 2(a–d) is the result from the GA solution to the sediment-trapping BMP placement optimization. The top image in each figure represents the solution found using linear programming. While the GA optimization required approximately 12 days of computing time on a desktop computer, the linear programming solution shown in the top image in each of Figs. 2(a–d) was found nearly instantaneously using *GAMS*. The linear programming solution captures some of the solution structure found with the GA.

Dynamic Programming Optimization of Sediment-Trapping BMP Placement

The results of sediment-trapping optimizations indicate that priority is given to placing BMPs close to the stream channel. This is likely because a sediment-trapping BMP can capture load that arrives from upslope, as well as load generated in the cell where it is placed. For reduction of sediment along a slope line, it is therefore never worse to place a fixed fraction removal-type BMP closer to the stream rather than farther away.

A dynamic programming approach to the sediment-trapping BMP placement problem is formulated based on the assumption that along a slope line of connected land parcels modeled as cells, the best location to place a BMP is the most-downslope cell. Similarly, the best location for the placement of the nth BMP along the same slope line would be in the nth most-downslope cell. Under these assumptions optimal BMP placement begins at the stream bank, a directly connected land cell in the distributed sediment model, and BMPs are added sequentially to cells in the upslope direction. The reduction in sediment load in a contributing area (k) is approximated with the following expression:

$$L_k = l_k(1 - r)^{p_k} \tag{14}$$

where L_k = load entering the stream from a directly connected watershed cell; l_k = sum of modeled load exported by the contributing area (k) to the directly connected cell in the absence of BMPs; r = fraction reduction achieved by the installation of a single BMP; and (p_k) = number of BMPs installed in contributing area k. The

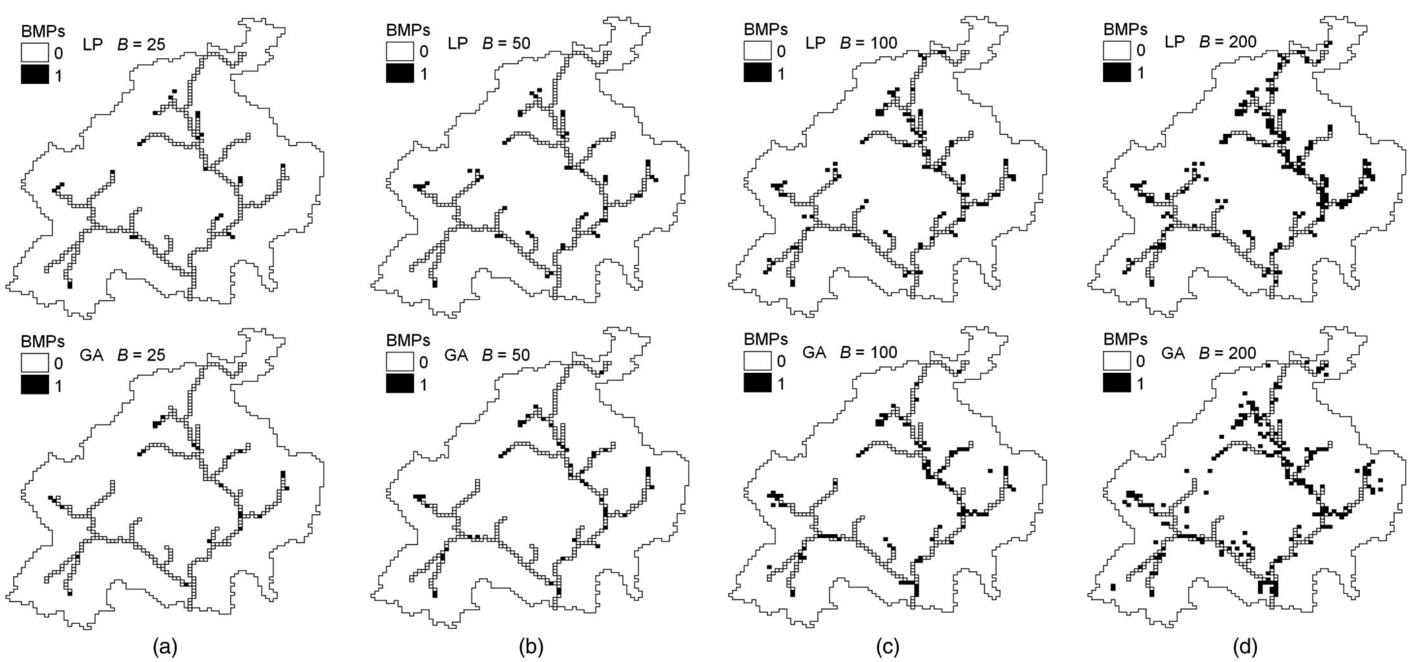

Fig. 2. Comparison of linear programming and genetic algorithm sediment load reduction optimizations: (a) $B = 25$; (b) $B = 50$; (c) $B = 100$; (d) $B = 200$

use of the index k is intended to distinguish the contributing area from the set of watershed cells (i). Each directly connected watershed cell defines a unique contributing area to the stream. The optimization is written as the minimization of total load (L_T), which is the sum of loads exported from all contributing areas (k), expressed as

$$L_T = \sum_k l_k (1-r)^{p_k} \tag{15}$$

subject to the budgetary constraint that

$$\sum_k p_k \leq B \tag{16}$$

where B = budget, expressed as the total number of BMPs allowed in the watershed.

Each contributing area exports sediment to the stream in parallel to all other contributing areas, p_k is an integer, and r is an assumed constant reduction fraction. Implementation of a BMP in Eq. (15) is represented by a fixed-fraction reduction in remaining sediment load from contributing area k. This is contrasted with the formulation in Eq. (9), where a BMP implementation results in a fixed-quantity reduction of each of the upslope loads. In each case, the goal is to find good locations for BMP implementation. More accurate quantification of load reduction could be achieved by subsequently running the optimized decision in a simulation model.

The problem in Eq. (15) is conceptualized as a dynamic program to implement the next BMP in a location that will yield maximum sediment reduction. At each stage, the next decision is based on results obtained by previously placed BMPs. A greedy algorithm can be used to quickly find the solution. The solution is generated by placing the first BMP in the contributing area with the largest load, l_k. The load of that contributing area is then updated by a factor of $1 - r$. Loads are then reranked to determine the contributing area with the new largest load, and the next BMP is placed in that location. This process is continued until the total number of BMPs reaches the budget, B.

A comparison of the optimization results obtained with the GA and those found using dynamic programming [Figs. 3(a–d)] shows close agreement between the two. The bottom image in each of Figs. 3(a–d) is the result from the GA solution to the sediment-trapping BMP placement optimization. The top image in each figure presents the solution found using dynamic programming and a removal fraction of $r = 0.5$. Shaded cells are locations of directly connected watershed cells where the greedy algorithm suggests placing one or more BMPs to remove sediment load entering the stream from the attached contributing area. The shading represents the number of BMPs applied, p_k from Eq. (15), with darker shades of gray indicating more BMPs.

As shown in Fig. 3(d), there is very close agreement throughout the riparian areas in the structure of the BMP solution generated by the dynamic program and the GA. The difference in the results between the dynamic programming and the linear programming solutions suggests the absence of an important mechanism in the formulation of the linear program. It is likely that the missing mechanism is natural sediment trapping.

The input to the linear programming model was the sum of modeled sediment generated in each cell of the watershed. It appears the linear program is responding to areas where there is a large generation of sediment load, and while this represents part of the solution, it neglects natural sediment trapping. In the nonlinear model described in Limbrunner et al. (2013), generated sediment was subject to potential natural deposition in downslope cells. The input to the dynamic program described earlier accounted for natural sediment trapping since the quantity of interest was the sum of modeled sediment exiting each directly connected watershed cell that drained contributing area k. Included in this sum are the effects of natural sediment trapping in the upslope contributing area. Natural deposition is independent of BMP placement but affects locations for optimal BMP strategies. The dynamic programming approach, which accounts for natural sediment trapping in the input load, performs better and more closely approximates the result of the GA optimization.

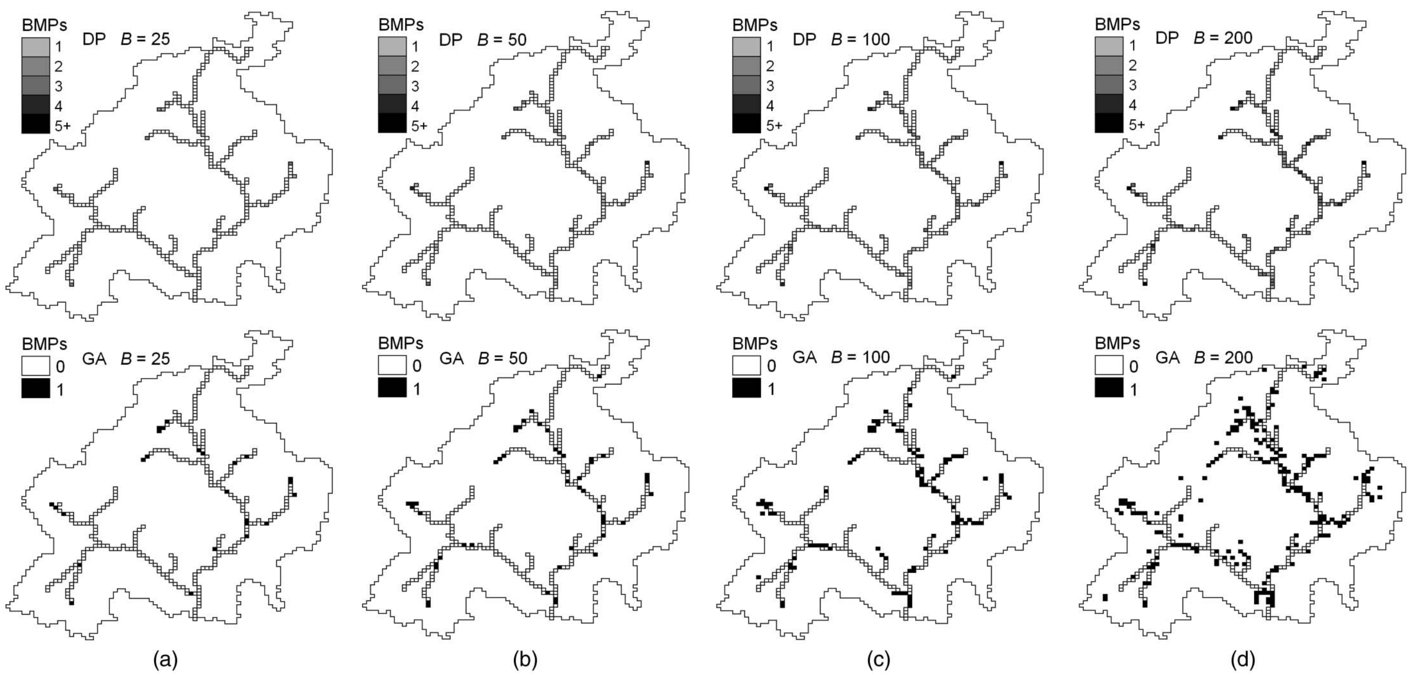

Fig. 3. Comparison of dynamic programming and genetic algorithm sediment load reduction optimizations: (a) $B = 25$; (b) $B = 50$; (c) $B = 100$; (d) $B = 200$

J. Water Resour. Plann. Manage. 2013.139:486-491.

Dynamic programming could be used as a screening tool to quickly reduce the search space that would be used with a GA, thereby reducing the time needed to converge on an optimal solution.

Conclusion

This study evaluated the use of classic optimization methods of linear and dynamic programming for their application to the optimization of stormwater and nonpoint source pollution management strategies. These methods generated solutions that compared well with solutions generated by the contemporary approach of using a GA with a complex distributed watershed model. The infiltration-based BMP optimization linear program introduced here appears effective at approximating the solution structure found using a GA and a nonlinear distributed model in Perez-Pedini et al. (2005). Similarly, the dynamic programming formulation introduced here can be efficiently solved to approximate the solution structure of the nonlinear sediment-trapping BMP optimization found with a GA in Limbrunner et al. (2013).

Complex simulation models have proven to be valuable tools for exploring stormwater and nonpoint source pollution management design. The combined use of simulation models and contemporary optimization algorithms has been extremely useful in dealing with these problems; however, the often lengthy computation times needed for optimization place a limit on the practicality of using these techniques in collaborative decision-making forums, such as stakeholder workshops. Classic optimization techniques provide both comprehensive and fast optimization capabilities that could help to overcome this limitation. Because of this, classic optimization techniques should be considered important tools to be used in conjunction with other techniques for developing collaborative decisions regarding stormwater and nonpoint source pollution management. Furthermore, both this study and that of Zoltay et al. (2010) emphasize that careful attention to the structural formulation of complex watershed management modeling can provide a useful alternative to the more computationally complex application of evolutionary algorithms and distributed nonlinear watershed models.

Acknowledgments

We would like to thank Professor Stephen Levine of Tufts University for his helpful insights. The work described here was funded in large part by a U.S. EPA STAR grant (R830654). We are grateful for the research opportunity the grant has provided. Since this work has not received the U.S. EPA's required peer and policy review, it does not necessarily reflect their views, and no endorsement should be inferred.

References

GAMS [Computer software]. GAMS Development Corporation, Washington, DC.

Jain, M. K., Kothyari, U. C., and Raju, K. G. (2005). "GIS based distributed model for soil erosion and rate of sediment outflow from catchments." *J. Hydraul. Eng.*, 131(9), 755–769.

Jenq, T.-R., Uchrin, C. G., Granstrom, M. L., and Hsueh, S.-F. (1983). "A linear program model for point-nonpoint source control decisions: Theoretical development." *Ecol. Modell.*, 19(4), 249–262.

Limbrunner, J. F., Vogel, R. M., Chapra, S. C., and Kirshen, P. H. (2013). "Optimal location of sediment-trapping best management practices for nonpoint source load management." *J. Water Resour. Plann. Manage.*, 139(5), 478–485.

Mays, L. W., and Bedient, P. B. (1982). "Model for optimal size and location of detention." *J. Water Resour. Plann. Manage. Div.*, 108 (WR3), 270–285.

Moglen, G. E. (2000). "Effect of orientation of spatially distributed curve numbers in runoff calculations." *J. Am. Water Resour. Assoc.*, 36(6), 1391–1400.

National Research Council. (2001). *Assessing the TMDL approach to water quality management*, National Academies Press, Washington, DC.

Perez-Pedini, C., Limbrunner, J. F., and Vogel, R. M. (2005). "Optimal location of infiltration-based best management practices for storm water management." *J. Water Resour. Plann. Manage.*, 131(6), 441–448.

ReVelle, C. S., Loucks, D. P., and Lynn, W. R. (1968). "Linear programming applied to water quality management." *Water Resour. Res.*, 4(1), 1–9.

Sample, D. J., Heaney, J. P., Wright, L. T., and Koustas, R. (2001). "Geographic information systems, decision support systems, and urban storm-water management." *J. Water Resour. Plann. Manage.*, 127(3), 155–161.

Schleich, J., and White, D. (1997). "Cost minimization of nutrient reduction in watershed management using linear programming." *J. Am. Water Resour. Assoc.*, 33(1), 135–142.

USDA. (1986). *Urban hydrology for small watersheds, technical release 55*, Soil Conservation Service, Washington, DC.

Zoltay, V. I., Vogel, R. M., Kirshen, P. H., and Westphal, K. S. (2010). "Integrated watershed management modeling: A generic optimization model applied to the Ipswich River Basin." *J. Water Resour. Plann. Manage.*, 136(5), 566–575.

J. Water Resour. Plann. Manage. 2013.139:486-491.