Multi-objective Cellular Automata Optimization

Epaminondas Sidiropoulos

Faculty of Engineering, Aristotle University of Thessaloniki, Greece
nontas@topo.auth.gr

Abstract. The role of cellular automata in optimization is a current area of research. This paper presents a multi-objective approach to cellular optimization. A typical nonlinear problem of spatial resource allocation is treated by two alternative methods. The first one is based on a specially designed operative genetic algorithm and the second one on a hybrid annealing – genetic procedure. Pareto front approximations are computed by the two methods and also by a non-cellular version of the second approach. The better performance of the cellular methods is demonstrated and questions for further research are discussed.

Keywords: Cellular automata, multi-objective optimization, genetic algorithm, simulated annealing, resource allocation.

1 Introduction

Cellular automata are being used for the simulation of a great variety of phenomena encompassing both natural processes and evolution of anthropogenic systems. In all those cases the system to be modeled is represented as a set or a lattice of discrete cells, to each one of which a certain "state" is assigned, coming from a set of possible states. The states of the individual cells are evolved according to a local transition rule. The local rule determines the next state of the cell as a function of the current states of the cells that are confined to the neighborhood of the cell in question.

The formulation of the transition rule is the key issue in modeling the evolution of a system. A suitable transition rule is determined, such that the produced overall simulation will be satisfactory, according to preset criteria. This determination can be achieved by means of evolutionary methods, such as genetic algorithms. A multitude of such simulations have been presented ranging from physical to social and economic, such as forest fire propagation [1], urban development [2], adsorption - diffusion processes [3], traffic flow [4].

An alternative view of the cellular automaton concept is its use and role in optimization. Indeed, cellular automata can play a significant role as conceptual tools in optimization, if the system to be optimized can be modeled as a discrete set of cells with a well defined neighborhood structure and with certain properties, that can be identified as states. Then, suitable local rules have to be found, such that, starting from an arbitrary initial configuration, the system will be guided toward optimal arrangements. Clearly, in this case the functioning of the transition rule will not be to

G.C. Sirakoulis and S. Bandini (Eds.): ACRI 2012, LNCS 7495, pp. 131–140, 2012.

produce a simulation, but to act as an adaptive local operator in an iterative procedure that aims at the optimum.

The concept of cellular automaton has been utilized in order to deal with a problem of groundwater management by Sidiropoulos and Tolikas [5]. Two alternative approaches were followed for that purpose: First, a genetic algorithm was embedded into a cellular automaton in order to effect the desired optimization. The chromosomes of the genetic algorithm reflected directly the local transition rules of the cellular automaton and an efficient solution was demonstrated. The second approach consisted in the application of simulated annealing. The perturbation involved in simulated annealing was chosen to take place inside the neighborhoods of the individual cells, thus forming the requisite local transition rule.

An idea similar to the above genetic algorithm was later applied to a problem of spatial groundwater allocation [6]. The typical chromosome expressed the local transition rule of the cellular automaton and thus it acted as an operator transforming a base configuration. Hence, the genetic algorithm presented was characterized as an operative genetic algorithm [7]. The same process was generalized and reinforced with local search by Sidiropoulos [8].

The simulated annealing approach described above was applied by Sidiropoulos [9] to a similar spatial resource allocation problem. Both an annealing and a mixed annealing - genetic approach were presented, always on a cellular background. In all cases the introduction of the cellular automaton mechanism and concept contributed to more efficient numerical procedures in addition to a more appealing and realistic representation of the problem.

This paper presents multi-objective versions of the above spatial groundwater allocation problems. The cellular automaton concept plays the central role in the proposed algorithms. The genetic algorithm of the previous cited works is extended to a multi-objective version retaining the operative character of the typical chromosome and incorporating the basic characteristics of Pareto front development. On the other hand, an extension of the simulated annealing approach is given in the form of a hybrid annealing – genetic algorithm with the local transition rule activating the perturbation of the annealing method. The same approach without the local aspect in the perturbation is shown to produce inferior results.

2 Problem Formulation

The single-objective versions of the present problem have been presented elsewhere ([6] and [9]). The problem is briefly reviewed here. A two-dimensional terrain is divided into land blocks as shown in Figure 1. Each one of the blocks is occupied by a certain cultivation and it receives irrigation water form one of three wells placed as shown in the same Figure. The land blocks can be considered to be the cells of the cellular automaton and the state of each cell will be identified with the well from which the cell (block) is irrigated. Each block is assumed to receive water from only one well and the set of possible states coincides with the set of the existing wells.

Contrary to the above references, two separate objective functions will be considered: f_p denoting the cost of pumping water at the sites of the wells and f_T denoting the cost of transporting water from the wells to the individual cells.

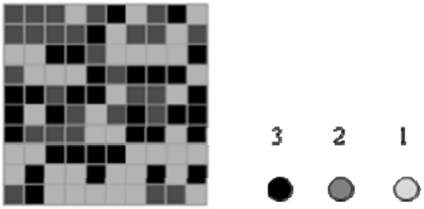

Fig. 1. Problem definition. The color of each cell signifies the well (1,2 or 3) to which the cell is connected

The cells are numbered consecutively according to the scheme presented by Sidiropoulos and Fotakis [6].

Let $\mathbf{C}=\{1,2,...\ell\}$ be the set of cells numbered consecutively.

Let **W** = {P, Q, R,…} be the set of wells. Also, let w: $\mathcal{C} \to \mathcal{W}$ be a function assigning to each one of the cells the well to which it is connected, i.e. w(i) $\in \mathcal{W}$, $\mathrm{i}=1,2,...,\ell$. The individual values w(i) will be identified as the states of the cells i (i=1,2,.., ℓ). The function w will determine the whole configuration, which is defined as

$$\mathcal{L}=\{\mathrm{w}(1),\mathrm{w}(2),...,\mathrm{w}(\ell)\} \tag{1}$$

The two objective functions associated with each configuration are now denoted more precisely as

$$\mathrm{f}_\mathrm{P}[\mathcal{L}]\in\mathbb{R}^+ \text{ and } \mathrm{f}_\mathrm{T}[\mathcal{L}]\in\mathbb{R}^+ \tag{2}$$

Both these costs are to be considered as functionals mapping the set of all possible configurations to the set of positive reals. The values of the functional f_P are determined with the help of a groundwater model that governs the function of an underlying aquifer, while the values of f_T are computed on the basis of the distances of the cell-blocks from the wells. From the nature of these objective functions it can be seen that the problem is both non linear and non-separable with respect to individual cell contributions. The details of the physical model and of the objective functions have been given elsewhere ([6], [7]) and will not be repeated here.

The problem is to determine the configuration (1) that minimizes the pair of functionals (2). According to the multi-objective optimization methodology, the Pareto front of non-dominated solutions will be sought (e.g.[10]). For this purpose, two alternative algorithms will be presented in the following sections, bearing a cellular automaton mechanism as their basic ingredient.

3 Method of Solution

3.1 A Genetic Algorithm Approach

Genetic algorithms have been proved to be particularly advantageous for the management of multi-objective optimization problems. A single-objective version of the problem described in the previous section was presented by Sidiropoulos and Fotakis [6], along with demonstrating the superior performance of a cell-based genetic algorithm. A bi-objective version of that approach is given in this section.

Let $\mathcal{N}$(i) denote the neighborhood of cell i., i.e. the set of cells neighboring to cell i in the sense of von Neumann and let $n(i) \in \mathcal{N}(i)$. Then the set

$$\mathcal{O} = \{n(1), n(2), ..., n(\ell)\} \tag{3}$$

can be considered as an operator acting on the configuration $\mathcal{L}$ of Equation (1), as follows: Let $\overline{w}(i) = w(n(i))$. Then the new configuration

$$\overline{\mathcal{L}} = \{\overline{w}(1), \overline{w}(2), ..., \overline{w}(\ell)\} \tag{4}$$

may be thought of as the product of an operation of the set $\mathcal{O}$ of Equation (3) on the configuration $\mathcal{L}$ of Equation (1) and thus be denoted as

$$\overline{\mathcal{L}} = \mathcal{O} \otimes \mathcal{L} \tag{5}$$

$\mathcal{O}$ obviously represents the rule that, for each one of the cells, picks out one of their neighboring cells. The cell in question will then adopt the state of the neighbor thus selected. In order to determine rules that iteratively lead to “better” configurations a genetic algorithm will be employed. This genetic algorithm will be called an operative genetic algorithm (OGA) [7] and the set $\mathcal{O}$ of Equation (3) will be its typical chromosome.

An initial base configuration $\mathcal{L}$ is formed with a random selection of the cell states. A population

$$\mathcal{O}_i \text{ (i=1,2,…,N)} \tag{6}$$

of such chromosomes will be formed and an equal number of new configurations will result according to the operation of Equation (5):

$$\overline{\mathcal{L}}_i = \mathcal{O}_i \otimes \mathcal{L}, \text{ i=1,2,..,N} \tag{7}$$

Each one of the new configurations of Equation (7) will be evaluated according to Equation (2):

$$\mathbf{f}_i = \left(f_{Pi}, f_{Ti}\right) = \left(f_P[\overline{\mathcal{L}}_i], f_T[\overline{\mathcal{L}}_i]\right) \tag{8}$$

The pairs $\mathbf{f}_i$, i=1,2,..,N of Equation (8) are considered as points on the two-dimensional space of the objective functions and they will be used for the selection of the new base configuration. In the single-objective version of the problem, the new base configuration was selected as the one with the best value of the objective function. In the present multi-objective problem, the selection will be done on the basis of two criteria: domination and isolation.

More specifically, from the set of points

$$\mathcal{F}=\{\mathbf{f}_i \mid i=1,2,..,N\} \tag{9}$$

of Equation (8), the non-dominated ones are separated. The concept of domination is defined in textbooks on multi-objective optimization (e.g.[10]). The notation $i \prec j$ is used to indicate that a point i dominates a point j, where $i, j \in \{1,2,..,N\}$. Thus the subset $\mathcal{F}_0$ of $\mathcal{F}$ that contains the non-dominated points can be written as

$$\mathcal{F}_0 = \left\{ \mathbf{f}_j \in \mathcal{F} \mid \neg\exists \quad i \in (1,2,..,N\} \quad \text{with} \quad i \prec j \right\} \tag{10}$$

On the other hand, for each one of the $\mathbf{f}_j$ pairs belonging to the set $\mathcal{F}_0$ of Equation (10), the distances d_j are formed from all other points of the set $\mathcal{F}$ of Equation (9):

$$d_{ji} = \left[(f_{Pi} - f_{Pj})^2 + (f_{Ti} - f_{Tj})^2 \right]^{1/2} \tag{11}$$

where j and i are such that $\mathbf{f}_j \in \mathcal{F}_0$ and $\mathbf{f}_i \in \mathcal{F}$.

From the above distances the smallest is chosen:

$$d_j = \min_i \left\{ d_{ji} \right\} \tag{12}$$

The distance d_j of Equation (12) is the distance of point j to the one nearest to it, on the objective space. Finally, among the non-dominated points, the one is found with the largest d_j:

$$d_{\text{base}} = \max_j \left\{ d_j \right\} \tag{13}$$

The point of the objective space with the index "base" of Equation (13) is characterized by a relative isolation with respect to the other points of the non-dominated set $\mathcal{F}_0$. Among the configurations $\mathcal{L}_i$ of Equation (7), the one with the index "base" will now replace the old base configuration $\mathcal{L}$:

$$\mathcal{L} \leftarrow \bar{\mathcal{L}}_{\text{base}} \tag{14}$$

The population of operators of the type given by (3), is the current population of chromosomes for the genetic algorithm. Therefore, they will now be subjected to the genetic operations of selection, crossover and mutation. Tournament selection will be adopted for this problem. Two points i and j are picked at random from the whole population. If one of them dominates the other, then the dominant is selected. Otherwise the distances d_i and d_j are considered. If $d_i > d_j$, then i is selected. Otherwise j is selected.

Crossover and mutation are executed in the standard fashion on the chromosomes of the type (3) and finally a renewed population is produced, whose members act as operators on the base configuration that was defined above. It needs to be noted here that distance was found to be a convenient way of characterizing isolation in the present approach. Other alternatives can be found in the literature, such as sharing distance [11] and crowding distance [10]

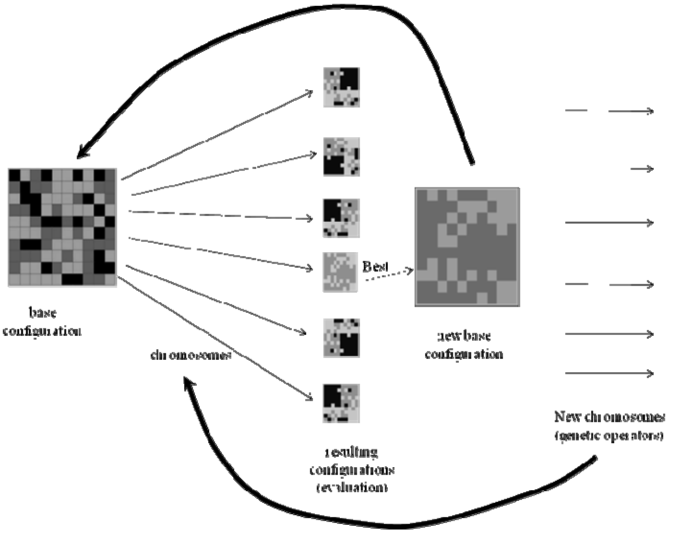
base configuration
chromosomes
Best
new base configuration
resulting configurations (evaluation)
New chromosomes (genetic operators)

Fig. 2. Outline of the multi-objective algorithm. The adjective best in front of the new base configuration means superior with respect to the criteria of domination and isolation as defined above.

The present algorithm is summarized as follows:

(a) An initial base configuration or mosaic is formed by randomly assigning a well to each one of the blocks.
(b) An initial population of operator-chromosomes is formed by randomly assigning to each block one of its neighbors.
(c) The chromosomes operate on the base configuration and generate an equal number of new configurations.
(d) Non-domination sorting is performed and the non-dominated configurations are kept separately.
(e) Out of all configurations of step (d) one is selected on the basis of domination and isolation and it becomes the new base configuration

(f) The current population of operator-chromosomes is subjected to tournament selection, crossover and mutation

(g) The renewed population of chromosomes of step (f) will operate on the base configuration of step (d) and new configurations are generated, just as in step (c).

(h) To the new configurations of step (g) the non-dominated ones of step (d) are added as elites.

(i) Control is transferred to step (d).

3.2 A Hybrid Annealing – Genetic Approach

An alternative approach to the genetic algorithm of the previous section is a scheme based on simulated annealing, but still retaining the characteristic of a population of solutions with a certain degree of genetic interaction. The concept of cellular automaton again plays a central role in the crucial step of perturbing the current base configuration. As with the operative genetic algorithm, there is also here a predominant base configuration, which does not stand separately, but it is a member of a population of configurations. This base configuration is perturbed, according to the annealing method.

In order for the resulting perturbed configuration to be subjected to the annealing test of acceptance, a certain value or fitness needs to be defined for a configuration. For multi-objective problems this is an open issue. In the present approach, the so called energy function is adopted for the evaluation of the possible solutions in multi-objective optimization [12]. The energy function is defined as follows:

```
For i=1,2,..,N
  $E_i$ = 1
  For j=1,2,..,N
    If $j \prec i$ then $E_i$ = $E_i$+1
    End If
  End For
End For
```

Extensive use of the concept of energy is made in reference [12], where more details are given. Below E_i denotes the energy of configuration $\mathcal{L}_i$:

$$E_i = E[\mathcal{L}_i]. \tag{15}$$

The above definition means that the lower the energy, the higher the value or fitness of the configuration. Thus the non-dominated points will have an energy value equal to 1. The proposed algorithm consists of the following steps:

(a) An initial population of configurations $\mathcal{P}= \{\mathcal{L}_i | i=1,2,..,N\}$ is formed according to the type given by Equation (1). These configurations give rise to a set $\mathcal{F}$ of N corresponding points (f_{Pi}, f_{Ti}) on the objective space.

(b) The initial base configuration is chosen at random from the N configurations.

(c) The current base configuration is set equal to the initial one of step (b).

The annealing double loop starts at this step:

(d) The current base configuration $\mathcal{L}$ is perturbed and a new configuration $\mathcal{L}_1$ results.

(e) The respective energies $E[\mathcal{L}]$ and $E[\mathcal{L}_1]$ are computed as above.

(f) Let $\delta E = E[\mathcal{L}_1] - E[\mathcal{L}]$.

```
If $\delta E < 0$, then $\mathcal{L}_1$ is accepted and $\mathcal{L} \leftarrow \mathcal{L}_1$
    Else    let $r$ = Random(0,1) and $p = \exp(-\delta E/T)$
            If $r < p$ then $\mathcal{L}_1$ is accepted and $\mathcal{L} \leftarrow \mathcal{L}_1$
                Else $\mathcal{L}_1$ is not accepted and $\mathcal{L}$ remains
            End If
End If
```

(g) By means of domination sorting algorithms, the subset $\mathcal{P}_d \subset \mathcal{P}$ of points is found that are dominated by the current base configuration $\mathcal{L}$. These points will eventually be removed from the population. Let the number of these points be equal to ℓ_{dom}.

(h) A randomly chosen subset $\mathcal{P}_r \subset \mathcal{P}$ is formed with cardinality equal to ℓ_{dom}-1

(i) The base configuration $\mathcal{L}$ is recombined with each one of the configurations that generated the points of $\mathcal{P}_r$. The recombination consists in an ordinary crossover with a single random separator. Out of the three new individuals involved in the crossover (one member of $\mathcal{P}_r$ and two offspring) the one is kept with the smallest energy function value (Equation 15). Thus ℓ_{dom}-1 members are now available. The set thus resulting is denoted as $\mathcal{P}_c$

(j) Finally, the new population will be composed as $\mathcal{P} \leftarrow (\mathcal{P} \setminus \mathcal{P}_d) \cup \mathcal{P}_c \cup \{ \mathcal{L} \}$, thus retaining the same number of configurations.

The annealing loop is completed with step (j) and control is transferred to step (d).

The temperature T that appears in step (f) is decreased every time a specified number of iterations are completed.

The idea of removing from the population the members dominated by the current point is known from the literature (Smith et al., 2008). The way of replacing the removed elements is different in the present approach.

The concept of cellular automaton comes into the algorithm when perturbing the current base configuration in step (d). Perturbation is carried out in two different ways. According to the first option, every cell will exchange states with one of its neighbors in a von Neumann neighborhood (Figure 3a). This is in full accord with the concept of cellular automaton and the method may be called cellular simulated annealing (CSA). Another mode of perturbation consists in an exchange of states between the current cell and another one anywhere in the space covered by the cells (Figure 3b). This is clearly incompatible with the notion of cellular automaton and it will be interesting to compare its performance to that of the CSA.

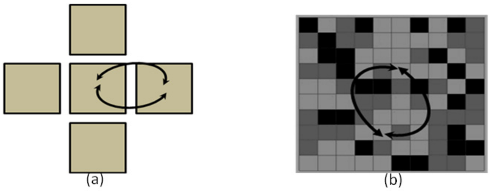

Fig. 3. Modes of perturbation

4 Results - Discussion

A fictive rectangular area was considered with an underlying aquifer of infinite extent. It was divided into 100 blocks and the specific data concerning the aquifer and the positions of the wells have been given in [6] and in [7]. As pointed out in the section on problem formulation, the problem is both nonlinear and non-separable. In previous works it was demonstrated that the operative genetic algorithm produced clearly superior results compared to more conventional, non-cellular approaches for the single-objective problem ([6] and [7]). Also, the application of simulated annealing on a cellular background gave results comparable to those of the operative genetic algorithm for the problem examined by Sidiropoulos and Tolikas [5] and clearly superior results compared to a more conventional version of simulated annealing for the problem treated by Sidiropoulos [9].

In the present multi-objective approach the operative genetic algorithm is compared to the CSA and to the non-cellular SA described in the previous section.

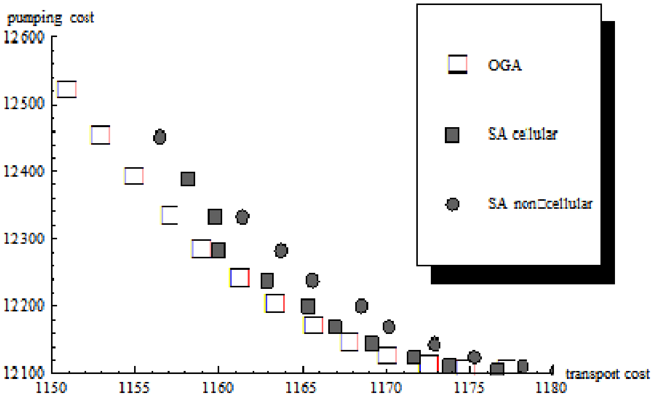

Fig. 4. Comparison of methods

All three methods produced approximations to the Pareto front. It is demonstrated in Figure 4 that OGA produced a clearly better front in comparison to the simulated annealing methods. The data depicted on the figure are not scaled, because classical scaling would have to be based on population maxima and minima. But in that case the results of the three different methods would not be directly comparable, as the respective populations could not have been the same. According to numerical experiments conducted by the author, the introduction of an elitism, as explained in the description of the algorithm given in Section 3.1, contributes to the performance of OGA. The cellular SA yielded a front close to that of the OGA, but not covering clearly the whole range of OGA.

Multi-objective cellular automata optimization has not been studied extensively as yet and a lot of research is needed regarding various methodological possibilities, as well as more extensive comparisons among methods. For instance, in the case of simulated annealing there are more alternatives to be considered with respect to the replacement of points removed due to being dominated by the current configuration.

References

1. Karafyllidis, I.: Design of a dedicated parallel processor for the prediction of forest fire spreading using cellular automata and genetic algorithms. Engineering Applications of Artificial Intelligence 17, 19–36 (2004)
2. Jennerette, G.D., Wu, J.: Analysis and simulation of land use change in the central Arizona – Phoenix region, USA. Landscape Ecology 16, 611–626 (2001)
3. Chopard, B., Droz, M., Kolb, M.: Cellular automata approach to non-equilibrium diffusion and gradient percolation. Journal of Physics, A: Mathematical and General 22, 1609–1619 (1989)
4. Salcido, A.: Equilibrium Properties of the Cellular Automata Models for traffic Flow in a Single Lane. Cellular Automata, Simplicity behind Complexity. In: Salcido, A. (ed.) INTECH (2011)
5. Sidiropoulos, E., Tolikas, P.: Genetic algorithms and cellular automata in aquifer management. Applied Mathematical Modelling 32(4), 617–640 (2008)
6. Sidiropoulos, E., Fotakis, D.: Cell-based genetic algorithm and simulated annealing for spatial groundwater allocation. WSEAS Transactions on Environment and Development 4(5) (2009)
7. Sidiropoulos, E., Fotakis, D.: Spatial optimization and resource allocation in a cellular automata framework. In: Salcido, A. (ed.) Cellular Automata: Simplicity Behind Complexity. INTECH Scientific Publishing (2011)
8. Sidiropoulos, E.: Cellular automata optimization via evolutionary methods. In: Li, T. (ed.) Cellular Automata. Nova Publishers (2011)
9. Sidiropoulos, E.: Spatial resource allocation via simulated annealing on a cellular automaton background. In: IEEE Proceedings of the World Congress on Engineering and Technology, October 28 –November. 2, vol. 2, p. 137 (2011)
10. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley (2001)
11. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization. In: Proceedings of the 5th International Conference on Genetic Algorithms, pp. 416–423 (1993)
12. Smith, K.I., Everson, R.M., Fieldsend, J.E., Murphy, C., Misra, R.: Dominance-Based Multiobjective Simulated Annealing. IEEE Transactions on Evolutionary Computation 12(3) (2008)