

Genetic Algorithms in Aquifer Management

Walter Cedeño and V. Rao Vemuri
Department of Applied Science
University of California, Davis and
Lawrence Livermore National Laboratory
Livermore, CA 94550
(wcedeno@llnl.gov and vemuri1@llnl.gov)

Abstract: A new genetic algorithm based on multi niche crowding is capable of efficiently locating all the peaks of a multi-modal function. By associating these peaks with the utility accrued from different sets of decision variables it is possible to extend the use of genetic algorithms to multi criteria decision making problems. This concept is applied to address the problems arising in the context of remediation of a contaminated aquifer. The multi niche crowding genetic algorithm is used to decide the optimal location of pumping wells. The aquifer dynamics are simulated by repeatedly solving the partial differential equations describing the flow of water using SUTRA code. Output of this simulation constitutes the input to the genetic algorithm.

Key words: Genetic algorithms, multi-modal functions, multi-objective optimization, aquifer remediation, ground water management.

1. Introduction

Many decision making problems of practical interest are characterized by multiple objectives. The decision maker is forced to strike a balance among many competing objectives, not all of which are quantifiable. The problem of simultaneously optimizing several objective functions has been posed and studied by many [Reid and Vemuri, '71; Vemuri, '74; de Neufville, '90]. What would be useful is an interactive method that presents to the analyst the consequences of choosing one of many competing alternatives. In this paper we intend to demonstrate the suitability of a new version of a genetic algorithm to address this problem. The crux of the method lies in identifying the objectives with the peaks of a function and solving for the locations and heights of these peaks using the new genetic algorithm.

Specifically, we study the remediation of groundwater in contaminated aquifers. The problem is to determine the optimum placement of pumping (and recharge) wells and optimum pumping (and recharge) schedules in order to achieve a set of objectives. Posing the remediation problem as a multi-objective optimization problem and solving that using genetic algorithms has been done [Rogers and Dowla, '94; Rogers, Dowla and Johnson, '94; Johnson and Rogers, '94; Horn and Nafpliotis, '93]. In this paper we address the same type of problem, but we use a new method, called multi-niche crowding GA [Cedeño and Vemuri, '92, '94; Vemuri and Cedeño, 94]. The problem is posed as one of locating the peaks of a function where the height of each peak in a $(n+1)$ -dimensional space corresponds to the value of a suitably defined utility function and the n coordinates of each peak correspond to the values of the n decision variables associated with that utility function.

2. The Simulation Environment

To lend reality and at the same time to keep the computational burden within practical bounds we studied a hypothetical groundwater basin beneath a one square mile region. Any resemblance this may have to a real aquifer is purely coincidental. Many simplifying assumptions about the flow of water are also made. The main objective here is to test the new genetic algorithm in an environment that is as realistic as it can be.

2.1 The Aquifer

Consider a 1 mile x 1 mile square region occupied by, say, a petrochemical facility. This site, by its very nature is likely to be polluted with spills of oil or other hydrocarbons. Such pollutants would seep into the ground and contaminate water in the aquifer, located, say, 90 to 180 feet below the ground. The primary concern is the containment of the pollutant from seeping outside the boundaries of the facility and polluting the drinking water supplies of the adjacent city which uses groundwater for municipal water supply. It is essential that contaminants from the site do not diffuse and pollute the city water supply.

In one operational scenario, polluted water is first pumped out, treated, and the treated water is used to recharge the aquifer using injection wells. As pumping and injection are expensive, the total cost (fixed and running) is often used as an objective function in the optimization procedure. The total volume of water pumped could well serve as an objective function also. Another possible objective would be the amount of contaminant actually removed. How does one go about designing the pumping network and pumping schedule to meet several objectives of this kind?

In our simulation we assumed that three "pump, treat and recharge " facilities already exist on the western edge of the site and one on the eastern edge. Three more "pump, treat and recharge" facilities are planned. The present task is to find an optimum location for these three new facilities. On the average each facility is assumed to cost \$2.5 million and is capable of cleaning 70 gallons of water per minute by reducing the concentration from 550 ppb to negligible quantities. Thus the scenario described is as realistic as we could make it. With proper modifications, this can be applied to a variety of other contexts.

2.2 The SUTRA Model

Flow of water in underground formations can be described by nonlinear partial differential equations (PDEs). Nonlinearities enter the equations because the transmissibility and storage coefficient, the two most important parameters, are dependent upon the elevation of the water table, the dependent variable in the PDEs [Vemuri and Karplus, '69]. The aquifers are non homogeneous and the flow could be two-phase. In the remediation context, solute mass transport also becomes important. The success of the management policies depend to a large extent on our ability to predict the movement of water and the contaminants over time. To meet these potential needs, we decided to use the flow and transport simulation code, SUTRA, of the U. S. Geological Survey [Voss, '84]. SUTRA (Saturated, Unsaturated TRANsport) is a 2-D hybrid finite-

element/ finite-difference model aimed at solving the governing partial differential equations for confined areal groundwater flow and areal solute transport.

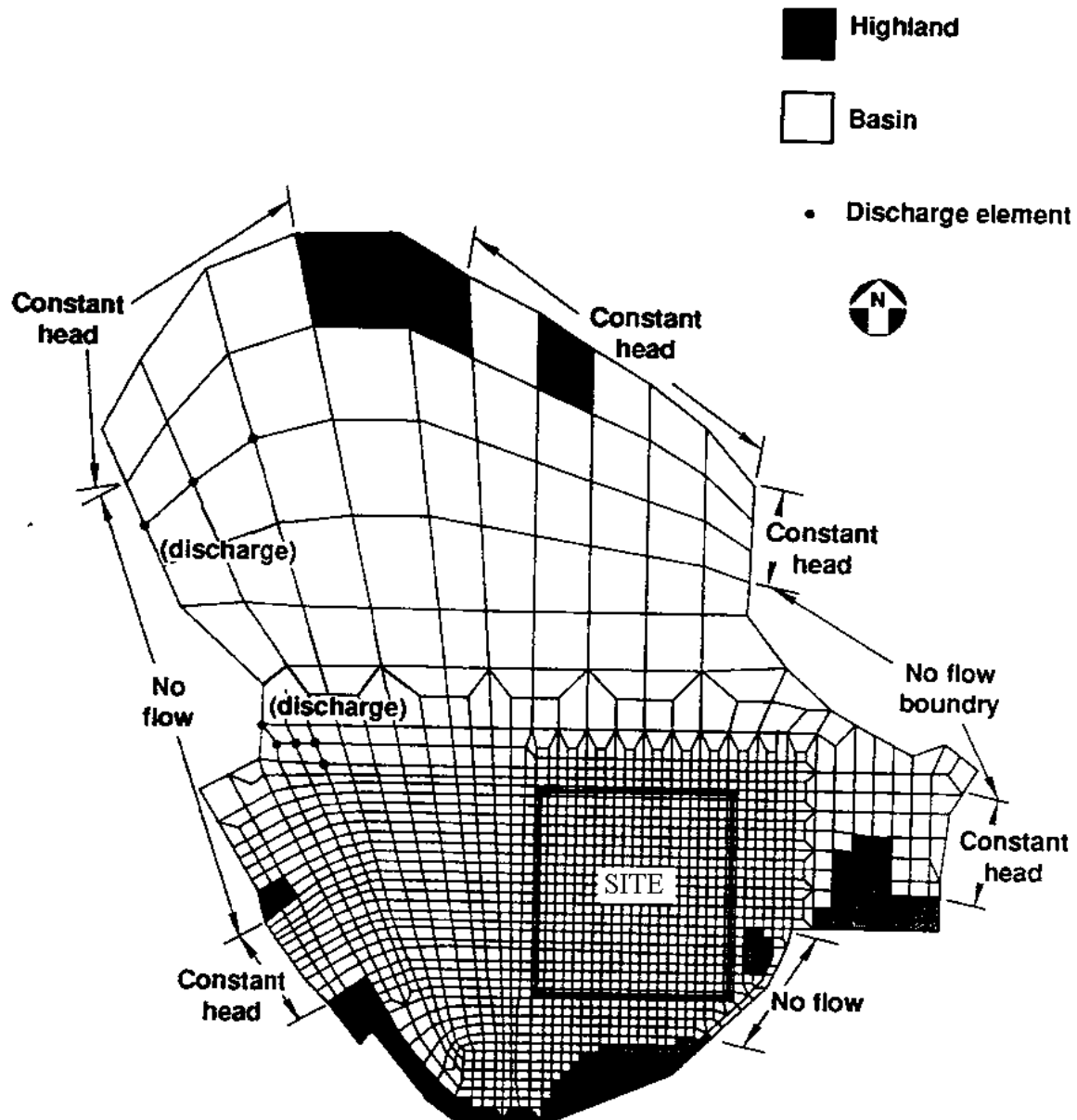


Figure 1: SUTRA nodes and elements in mesh

Boundary conditions imposed will depend upon the specific problem. In this hypothetical example, we assumed no-flow fault zones to the northeast and southeast. Flux boundaries were

assumed along the eastern and western edges of the site. A hydraulic conductivity of about 10 feet/day is assumed.

The mesh used has a total of 2436 nodes and 2385 elements, which covers the areal extent of the aquifer, which is much bigger than the one square mile region (see Figure 1). Out of this, we have chosen a subregion of size 20 x 21 nodes, covering an area of 5200 ft x 4950 ft. Each node in this subregion is considered as a potential pump location. The distance between nodes is 260 feet, and there are 420 possible pumping locations. Strictly speaking, the decision on where to locate the pumps should be governed by considerations such as the concentration of pollutants, pollution gradients, feasibility of drilling, and so on. In this study, we were forced to include another consideration, namely the computational burden. SUTRA is computationally intensive; it took about 6.5 minutes of elapsed time per fitness evaluation. Each fitness evaluation ran SUTRA for 10 time steps, where each step had a duration of one year. Any final resolution meant much more time and we simply could not muster the computational resources to do the job. It was felt that some gains in computational time in terms of fitness evaluations can be made by restricting the potential pumping sites to a subregion of the total grid and by using larger time steps.

This knowledge of the flow field, obtained by solving the PDE's using SUTRA, constitutes an input to the genetic algorithm. The algorithm's goal is to find sets of well locations in the 20 x 21 grid that best meets the objectives, subject to the constraint that no more than 10 wells are allowed. The algorithm outputs the values of the objective functions for a range of values of the decision variables. Each of the objective functions constitutes the "modes" of the multi-modal function over which the genetic algorithm conducts its search.

3. Proposed Genetic Algorithms for Multi-modal Search

In this paper the term *multi-modal function* is used to imply any function of several variables exhibiting more than one peak. Searching for the extrema in a multi-modal search space is different from locating the extremum of a uni-modal function. When a search technique, proven to be useful for uni-modal functions, is applied to multi-modal functions, the method tends to converge to an optimum in the local neighborhood of the first guess. One can use tricks like simulated annealing to escape from a local optimum and locate the global optimum. However, there are many applications where the location of "k best extrema" of a multi-modal function are of interest. Searching for these locations goes by the name "multi-modal optimization."

A genetic algorithm, GA for short, is a stochastic search technique based on the principles of biological evolution, natural selection and genetic recombination [Holland, '75; Goldberg, '89]. GAs simulate "survival of the fittest" in a population of solutions in a search space. GAs have been shown to work well in noisy environments and in complex search spaces. In GA parlance, a possible solution to a problem is referred to as an *individual*. An individual is represented by a computational data structure called a *chromosome*, which is encoded using a fixed length alphabet, usually a binary alphabet. A fitness function is designed such that fitness of individuals, or groups of individuals, moves toward an extremum if they carry some desirable traits. These desirable traits can be imparted to a new solution by selecting parents having those traits and mating them, through a mathematical operation, called *crossover*. A different region of the search space can be explored by generating a new solutions from an existing solution via a *mutation* operation. *Species* are individuals with a common characteristic and *niches* are sub domains of the search space. By encouraging niching and speciation, GAs can facilitate convergence to more than one extremum in a multi-modal search space.

There are many versions of genetic algorithms. In a nutshell, all genetic algorithms have three basic steps: selection, recombination (or, reproduction) and replacement. During the *selection step*, a decision is made as to who, in the population, is allowed to produce offspring. During the *recombination step*, offspring are produced via the operations of crossover and mutation. During the *replacement step* another decision is made as to which of the members from generation i are forced to perish (or vacate a slot) in order to make room for an offspring to compete (or, occupy a slot) in the next generation. Implicit in this replacement step is the assumption that the population remains constant (the steady state assumption) from generation to generation. Various versions of GA's differ from each other in the details of how these steps are implemented.

The Simple GA (or, SGA), which will be used as a point of departure for presenting the MNC GA, starts by randomly generating a population of N individuals. These individuals are evaluated for their fitness. Individuals with higher fitness scores are selected, with replacement, to create a mating pool of size N . Because the selection is "with replacement" an individual with more than average fitness score can get more than one chance to participate in the reproduction step. This method of selection goes by the name roulette wheel selection, or *fitness proportionate reproduction* (FPR). The crossover genetic operator is applied at this stage in a probabilistic manner which results in some individuals from the mating pool to actually reproduce. If the probability of crossover is set at 1.0, all members of the mating pool reproduce. For simplicity, it is assumed that each pair of parents produce only one pair of offspring through the crossover operation. The mutation genetic operator may be applied at this stage in a probabilistic manner so that some of the bits in the chromosomes of the offspring are randomly altered. Now the population pool contains some individuals who never got a chance to reproduce and the offspring of those who got a chance to reproduce. That is, the second generation is comprised of those first generation individuals who never got a chance to reproduce and the offspring of those who got a

chance to reproduce. The procedure continues until a suitable termination condition is satisfied. All other versions of GAs differ from this basic method in some detail or another.

The steady state GA, or SSGA (which in our opinion is a misnomer) differs from SGA mainly in the replacement step, and to a lesser extent on the way the genetic operators are applied (Whitley '88; Syswerda '89). The SSGA selects two individuals using FPR and allows them to mate to produce two offspring. This selection step is identical to the corresponding step of SGA. However, in SSGA, the offspring are inserted into the population, thus replacing two individuals, soon after they are generated whereas the SGA generates N offspring prior to replacing the entire population. In other words SGA uses *simultaneous* replacement strategy whereas the SSGA uses the *successive* replacement strategy. Thus SGA and SSGA are analogous, respectively, to the Jacobi and Gauss-Seidal methods of solving systems of linear algebraic equations [Vemuri and Karplus, 1981]. From the perspective of the size of the population, both the SGA and SSGA are steady state genetic algorithms, because the total population from generation to generation remains fixed. Therefore, the SGA is more appropriately called by the descriptive term, *GA with simultaneous replacement of populations* and the SSGA by the term, *GA with successive replacement of populations*.

Both SGA and SSGA suffer from the possibility of premature convergence to a local minimum, primarily due to the *selection pressure* exerted by the FPR rule. Once a candidate with an above average fitness score is identified, the FPR rule begins to favor that candidate from generation to generation, unless a better candidate emerges through crossovers and mutations. This means that the FPR simply assigns an exponential number of mating chances (this, in a sense is the Schema Theorem) to those members of the population that exhibit above average survival rates. This process is analogous to the process of convergence to a local minimum from an initial guess in the neighborhood of that minimum. Although this strategy is good for quickly locating a peak and converging toward it, it is not a good strategy for a thorough exploration of

complex search spaces with multiple peaks. Due to this reason as well as the *deceptiveness* exhibited by multi-modal search spaces (Goldberg et al, '92), SGA and SSGA are not suitable for multi-modal search and optimization.

3.1 Multi-modal Search: Crowding-based Methods:

If a search space is characterized by many peaks and if one is interested in finding the locations and heights of all (or, at least the k-tallest) peaks, then one has to develop a hybrid strategy that combines both exploitation of the results on hand as well as further exploration of the search space. The MNC GA described in this paper is one such hybrid strategy. Although thorough theoretical analysis of the MNC GA is not available at this time, initial experiments with the proposed method are promising. Experimentally we have shown that the MNC GA has the ability to converge to multiple solutions simultaneously. It achieves this result by encouraging competition between individuals within the same locally optimal group [Cedeño and Vemuri, '92] while limiting global competition to a minimum. This is what nations do when they wish to protect local industries from the onslaught of global competition. To achieve this goal, in MNC GA both the selection and replacement steps of the SGA are modified with the introduction of some form of *crowding* in order to render it suitable for searching spaces characterized by multiple peaks or niches.

Crowding [De Jong, '75] is a generalization of *preselection* [Cavicchio, '70]. In *crowding*, selection and reproduction are the same as in the SGA; but replacement is different. For concreteness, it is assumed that two parents are selected to produce two offspring. In order to make room for these offspring, it is necessary to identify two members of the population for replacement. The policy of replacing a member of the present generation by an offspring is carried out as follows, in two steps. First, a group of C individuals is selected at random from the population. C , called the *crowding factor*, indicates the size of the group. A value of $C = 2$ or 3

appears to work well for De Jong. Second, the bit strings in the chromosomes Offspring 1 are compared with those of the C individuals in the group using Hamming distance as a measure of similarity. The group member that is most similar to Offspring 1 is now replaced by the Offspring 1. This procedure is repeated for Offspring 2 as well. Note that Offspring 2 can conceivably replace its own sibling, namely Offspring 1, that just entered the population pool, although such a scenario is rather unlikely. In any event, crowding is essentially a successive replacement strategy. Summarizing, in crowding offspring replace similar individuals from the population. By doing this, crowding strategy strives to maintain the diversity in the population and postpones premature convergence. That is, crowding slows down premature convergence of the traditional GA, and in most cases can find the global optimum in a multi-modal search space. However, crowding cannot maintain stable sub populations because the FPR continues to exert selection pressure.

In multi-niche crowding (MNC), both the selection and replacement steps are modified with some type of crowding. The idea is to eliminate the selection pressure caused by FPR while allowing the population to maintain some diversity. This objective is achieved, in part, by encouraging mating and replacement within members of the same niche while allowing for some competition for population slots among the niches. The result is an algorithm that (a) maintains stable sub populations within different niches, (b) maintains diversity throughout the search, and (c) converges to different local minima.

In MNC, the FPR selection is replaced by what we call *crowding selection*. In crowding selection, each individual in the population has the same chance for mating in every generation. Application of this selection rule takes place in two steps. First, an individual A is selected for mating. This selection can be either sequential or random. Second, its mate M is selected, not from the entire population, but from a group of individuals of size c_s , picked at random from the

population. The mate M thus chosen must be the one who is the most "similar" to A . The similarity metric used here is not a *genotypic metric* such as the Hamming distance, but a suitably defined *phenotypic distance metric*. Crowding Selection promotes mating between individuals from the same niche while allowing matings between individuals from different niches.

During the replacement step, MNC uses a replacement policy called *worst among the most similar*. In a sense, this is a minmax strategy. The goal of this step is to pick an individual from the population for replacement by an offspring. Implementation of this policy follows these steps. First, c_f groups are created by randomly picking s individuals per group from the population. These groups are called *crowding factor groups*. Second, one individual from each group that is most similar (maximally similar) to the offspring is identified. This gives c_f individuals that are candidates for replacement by virtue of their similarity to the offspring that will replace them. From this group of most similar individuals, we pick the one with the lowest (minimum) *fitness* to die and that slot is filled with the offspring.

Figure 2 contains an example of worst among most similar replacement. The following pseudo-code summarizes the salient features of the method:

1. Generate initial population of N individuals
2. For $gen = 1$ to MAX_GEN
3. For $i = 1$ to N
4. Use *crowding selection* to find mate for individual i
5. Mate and mutate
6. Insert offspring in the population using *worst among most similar* replacement.

If we use FPR in Step 4 shown above and replace the lowest fitted individuals in the population (Step 6) with the newly generated offspring, this model corresponds to a steady-state GA. In contrast with the most common generational GA, offspring are available for mating as soon as

they are generated, and good individuals can survive for many generations. For the purposes of this paper, a generation is every N mating operations, where N is the population size.

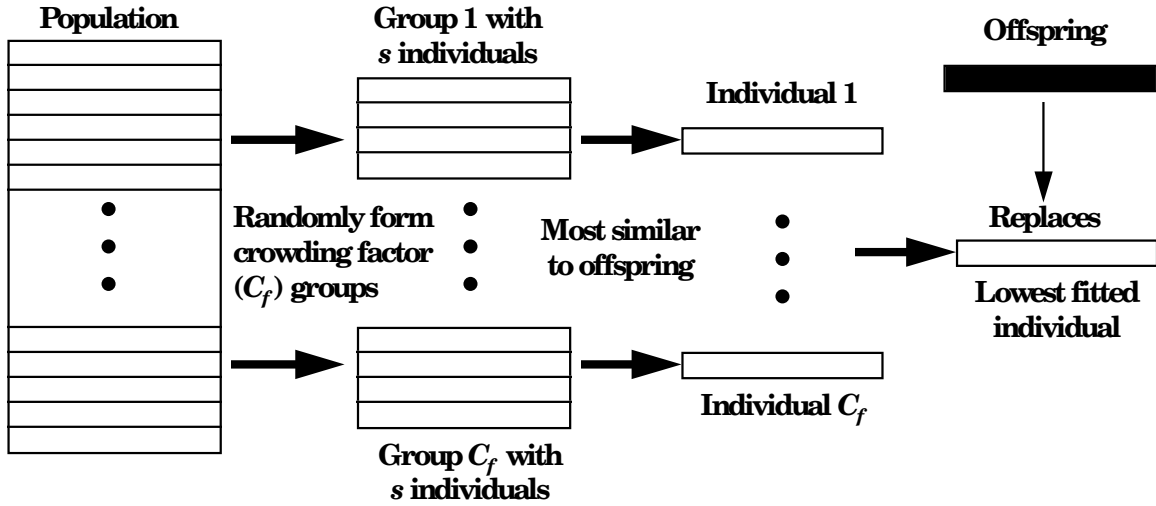


Figure 2: Schematic showing crowding factor groups created during the replacement step.

4. Interfacing with the Genetic Algorithm

In this section we describe the genetic operators for the aquifer problem. Specifically, the problem being solved is the determination of the optimum location of no more than 10 wells, on a 20 x 21 grid, so that three objectives are met. The first objective is to minimize the remediation cost which includes the cost for facilities, piping, water treatment, and day to day operation. Cost minimization can be achieved by setting up a budget and trying to find solutions which stays within or close to the budget. This approach allows the GA to treat solutions that are within the budget equally, and alleviates convergence pressure to single-well solutions. The second objective is to maximize the amount of contaminant removed from the aquifer. This is a straightforward measure and is obtained from the output of SUTRA in kilograms. The third objective is to prevent unsafe levels of contaminant from leaving the site. The goal is to minimize the concentration of contaminant leaving the site as much as possible. This measure is also obtained from the output of SUTRA in parts per billion (ppb).

4.1 Chromosome Representation

A solution to the problem is represented by a dynamic-length chromosome, i.e., a chromosome whose length can change. Each chromosome is comprised of information on up to 10 wells, where the well locations appear in ascending order. The well locations are identified by node numbers in the 20 x 21 grid delineating the site in the SUTRA mesh. No duplicate well locations are allowed in the chromosome. For example, the chromosome (41 107 224 635 972) denotes a five-well pumping scenario with the wells located at the nodes 41, 107, 224, 635, and 972. Only 420 of the total 2436 nodes in the mesh are valid well locations; wells are not feasible at other nodes because those nodes fall outside the 1 square mile site. This representation is equivalent to a binary string with a bit for each possible well location in the grid. In the future we intend to include attributed nodes in the chromosome. An attributed node includes not only the node number but also other relevant information about that node such as pumping rates, pollution removed, etc. This modification should render our representation more practical. This attributed representation can accommodate constraints on well locations due to the presence of buildings or other structures.

4.2 Recombination

During the recombination step, the genetic operators of crossover (or, mating) and mutation are applied. After two individuals in the population are selected using crowding selection the mating operator is applied to generate one offspring (or, child). The mating operator used here is equivalent to uniform crossover [Syswerda, '89]. A child chromosome is formed by first passing to it the genes that are common to both the parents and then selecting with a 0.5 probability the unique (i. e. non common) genes in both the parents. For example, let individual A = (41 107 224 635 972) and its mate M = (22 107 635 700). The child resulting from mating A and M will have wells at nodes 107 and 635 because these well locations are common in both the parents. The other well locations in the child will come from some subset of the remaining five non-common

nodes, {22, 41, 224, 700, 972}. These well locations are selected with probability 0.5. A resulting offspring can be (22 107 635 972).

Three mutation operators are considered: "add well", "delete well", and "move well". All three operators are applied independently with the same mutation probability. The offspring undergoes mutation if the flip of a biased coin is true. During an "add well" mutation a well in one of the 420 nodes in the grid is selected at random and added to the offspring while taking care to see that no duplicate nodes are allowed. The "delete well" mutation removes a well from the offspring at random, while taking care to see that only offspring containing more than one well can undergo "delete well" mutation. Finally, the "move well" mutation changes the location of a well, selected at random, to one of the four nodes around it. The new location is either at the top, bottom, left, or right of the previous location in the grid. The "add well" and "move well" mutations allow new well locations to appear that might not have in the initial population. The "delete well" mutation prevents the individuals from growing without control. The "move well" mutation allows some fine tuning.

4.3 Replacement

The replacement step in the MNC (worst among most similar) selects, for replacement, the individual in the crowding factor group with the lowest fitness. In a multiobjective problem a single fitness value can only be obtained if the objectives are grouped into a single utility function with weighting values. This is not appropriate since a specific set of weight values drives the GA towards specific solutions. Here we determine the worst individual in the group by ranking the individuals for each objective and selecting the individual with the worst total ranking. This is a type of minmax strategy where an individual with a minimum fitness from among a group of individuals who are maximally similar to a child is selected for replacement by that child. This concept is pictorially shown in Figure 3. The individuals in the crowding factor group, indicated by A, B, C, and D, are ranked for each objective. Here a lower rank number indicates a better value

for the objective being considered. Same rank numbers are assigned to individuals with equal or similar objective values. The total ranking for each individual is then determined by the sum of all the ranking numbers. The individual with the highest total rank, in this example B, is replaced by the offspring in the population.

Using ranking as a way to determine the worst individual in a group allows good individuals for specific objectives to evolve as well as individuals that are average in two or more objectives.

Individual	Objective 1	Objective 2	Objective 3	Total Ranking
A	2	1	3	6
B	2	4	2	8
C	1	2	3	6
D	3	3	1	7

Figure 3: Use of ranking in Worst Among Most Similar Replacement.

4.4 Similarity Measure

Similarity between individuals is used, as a distance measure, during the selection and replacement steps in the MNC GA. It allows formation of species within the population naturally and maintains diversity throughout the run. For this problem, a simple measure based on the number of wells in each individual chromosome and the distance between the closest well match is used as a similarity measure. For example, to calculate the distance between individual A = (a1 a2 a3 a4) and individual B = (b1 b2 b3 b4 b5 b6) with 4 and 6 wells respectively, the following three steps are implemented. First, the squared difference in the number of wells between the individuals is calculated. In this example, $(4 - 6)^2 = 4$. Second, the individual with the lower number of wells is selected, in this case A. Finally, for each well in A, the *closest* well in B is determined, say (a1,b4), (a2,b1), (a3,b6), and (a4,b3). The *similarity* between individuals A and B

is now given by the sum of squared difference and the distance between all pairs of wells. The following equation summarizes the distance formula for this example:

$$\text{similarity} = \text{sqrDiff} + \text{dist}(a1,b4) + \text{dist}(a2,b1) + \text{dist}(a3,b6) + \text{dist}(a4,b3).$$

This similarity measure indicates how similar two individuals are. The smaller the number the more similar the individuals are.

4.5 Fitness Measure

There is no explicit fitness function defined in this problem. The fitness of each individual is implicitly given by their objective values and their rank against other individuals. Survival of an individual from generation to generation is determined by competition among small groups of individuals during replacement. Specifically, an individual will survive for many generations if its total ranking, as described in Section 4.3, given by the objective values is lower than those individuals in the crowding factor group. For the aquifer problem we are using three objectives; cost, contaminant removed, and the ability to meet regulatory constraints.

Cost for a set of pumping realizations is calculated using the distance of each well to an hypothetical pump and treat facility located in center of the 20 x 21 grid. In the future we plan to include the location of this facility as part of the chromosome definition. The total distance is used to estimate the cost for piping, which is estimated as \$70 per ft [Rogers 92]. The cost objective is to find a set of wells which stays close to an established budget. In our runs we used a budget of \$2.5 million dollars. The squared difference between the budget and the piping cost is the value being minimized.

The second objective is straightforward. The amount of contaminant removed is taken from the estimates obtained from SUTRA runs. This value is output in kilograms and is dependent on the well locations and the amount of water being pumped. In this implementation we are using a fixed pumping rate of 0.0223 cubic feet per second for all wells. In the future we plan to include

the pumping rate as part of the chromosome. The goal in formulating this objective is to maximize the amount of contaminat removed.

The third objective is to meet the regulatory limits imposed. For this, a set of monitor wells are located on the west side of the site. The amount of contaminat in parts per billion (ppb) is estimated using SUTRA. We assumed that the total amount of contaminant reaching the monitor wells should be no more than 20 ppb. This value is given by SUTRA and the goal in this problem is to minimize the amount of contaminant as much as possible.

The values calculated for each objective are used by the MNC GA only during replacement step. Each objective value is used to determine the ranking of the individual for the particular objective. The sum of the rank values is then used to select the individual to be replaced by the offspring. Any number of objectives can be included in the problem. An objective which ranks an individual according to its similarity to members of the population can also be use to increase the diversity [Winston 1992] in the population.

5. Results and Discussion

The results described in this section were obtained on a Sparc 2 workstation running Sun OS 4.1.3. Following is the list of parameter values used for the MNC GA.:

Population size:	25
Crossover probability	1.0
Mutation probablity	0.1
Crowding selection size:	3
Crowding factor:	3
Crowding size:	3
Number of generations:	20, 25

In addition to restricting the potential pumping sites to a subregion of the total grid, small population sizes and a small number of generations were selected primarily to reduce the

computational burden. The crowding parameters were selected after trying some combinations in the interval [2-5]. From previous experience we had learned that values between 5% and 15% of the population size are reasonable for the crowding selection size and the crowding size parameters [Vemuri and Cedeño, 1994]. Crossover is always applied to mates, hence the crossover probability is set at 1.0. The mutation probability was set at 0.1 to allow on average 10% add, move, and delete mutations every generation.

The initial population was generated at random. For each individual the number of wells in the chromosome was chosen at random from the interval [1, 10]. Then, the well locations were selected at random from the 420 nodes in the 20 x 21 grid. In all runs the initial population never contained a solution meeting the regulatory limit, assumed for the sake of concreteness to be 20.0 ppb.

Scenario 1: Two Objective Functions

At first, the MNC GA was implemented to solve a two objective optimization problem. The first objective is to restrict the amount of contaminant leaving the site to a presumed regulatory limit of 20 ppb. The second objective is to maximize the contaminant removed from the pumped water at the treatment facilities. Although it may appear at first sight that achieving the second objective automatically results in the attainment of the first objective, it is not always necessarily so. Contaminant transport in ground water basins is very slow. If the pumping wells are located at some distance from the site boundary, it is conceivable that the effects of pumping will reach a far off location (say, the boundary of the site) only after several years. This in turn would mean that the SUTRA simulation should be conducted for long durations of time. This is one aspect of the computational bottleneck that prompted several simplifying assumptions in the numerical simulations.

Table 1: Sample solutions with two objectives

Gen.	No. of Wells	Contaminant leaving site (in ppb)	Contaminant Removed (in kg)
4	7	19.23	4.52
5	9	1.90	25.87
7	8	3.80	35.25
12	7	1.29	35.38
13	8	13.36	75.62
15	10	0.92	51.94
20	9	0.31	46.82

As the regulatory constraint is a dominating constraint, it was decided that only those solutions that satisfy this constraint are of interest. Table 1 shows a sample of these solutions. Notice the absence of generations 6, 8, 9, 10, 11, etc. In these generations acceptable solutions were found, but they didn't improved significantly over the solutions seen so far. Among the solutions not shown in the table, there are some solutions that did a very efficient job of removing the contaminant but did not meet the regulatory limit. This apparently anomalous behavior is probably due to the above mentioned transport delays between the pumping well locations and monitor well locations.

The acceptable solutions are then examined for their ability to remove the contaminant. The solutions are arranged in the order of generations in which they appeared for the first time. That is, in Generation 4 (first row) there appeared an individual (a solution), for the first time, that satisfied the 20.0 ppb regulatory limit, and that individual solution is characterized by seven pumping wells, and that seven-well pumping configuration managed to remove 4.52 kg of the contaminant, while the monitor wells recorded a 19.23 ppb concentration leaking out of the site under study.

Further examination of the table reveals that all solutions meeting the regulatory limit are characterized by the presence of seven or more pumping wells. Both the 7-well solutions, from Generations 4 and 12 are consistent with each other, although the solution appearing in the 12th generation is better. The difference between the 4th and 12th generation solutions comes presumably because of the actual placement of the seven wells in question. Similar behavior is observed with respect to the two 8-well solutions and the two 9-well solutions shown in the table.

Figure 4 shows the solutions represented by the fourth and fifth rows in Table 1. The locations of the wells in the map, denoted by symbols for hand pumps, are only approximate. These solutions have four wells in common although they have different objective values. In general small changes in well locations may cause large changes in the objective values. The MNC GA alleviates this behavior by promoting recombination between similar solutions. In both solutions the wells are mostly in the southern part of the site where the pollutant concentration is greater. Multiple wells in close proximity and wells in the north-east segment of the map can be considered useless and may be removed from the final configuration.

Scenario 2: Three Objective Functions

In this scenario we included the cost objective in the fitness function. The cost objective is to minimize the difference between the cost of installation and an established budget. Some positive changes were observed in the quality of solutions. The cost objective added some pressure towards smaller number of wells. Configurations with fewer wells were more common in later generations than before. At the same time the best solutions removed a greater amount of contaminant and had lower amounts of contaminant leaving the site. In general the MNC GA was able to improve the three objectives in subsequent generations.

Table 2: Sample solutions with three objectives

Gen.	No. of Wells	Contaminant leaving site (in ppb)	Contaminant Removed (in kg)	Cost Rank
10	7	6.30	36.71	6
10	6	7.79	39.62	5
18	7	11.15	43.37	4
19	7	13.27	59.02	3
20	7	2.44	64.68	1
21	7	1.21	81.98	2
23	9	6.59	98.89	7

Table 2 shows a set of solutions for a run of the MNC GA. Only improvements from previous generations are shown. Only solutions meeting the regulatory limit and having a higher amount of contaminant removed from previous generations are shown. The last column ranks the solutions according to their cost. Lower rank values indicate lower cost solution. As expected, improvements in one objective had a negative impact in other objectives during the initial generations. The competition by solutions having above average ranking in particular objectives was visible again and as a result offspring with good ranking in must objectives appeared in later generations.

Figure 5 shows the placement of the wells for solutions shown on the fifth and sixth rows of Table 2. Both solutions have seven wells with five of them common between them. Both solutions have very low values for the amount of contaminant leaving the site. They also have high rankings in cost and contaminant removed. Prior to what we had seen using only two objectives, improvements in more than one objective was more common in later generations. Better results may be obtained if we increase the number of generations executed and the population size.

6. Discussion and Conclusions

Several issues were addressed in this paper. First, the scope of this paper is limited to the issue of demonstrating the ability of the new GA, namely the multi-niche GA, in finding an acceptable solution to the multi-objective optimization problem arising in a practical context. Performance of MNC GA relative to alternate approaches is being investigated. Some preliminary results obtained when MNC GA is applied to artificially created, multi-modal test functions have already been reported [Cedeno and Vemuri, 95].

This paper clearly demonstrated the suitability of MNC GA in meeting this restricted scope. It is shown that the MNC GA indeed has the ability to maintain different solutions satisfying multiple (and perhaps, conflicting objectives). The stratagem of replacing a single utility function to measure fitness by a rank score assigned to various objectives appears to have allowed non-inferior solutions (i. e., solutions with favorable rankings in all objectives) to evolve. Given the limits imposed by us on the population size and the maximum number of generations in the evolutionary process, the quality of solutions obtained appear to be satisfactory. The policy of maintaining diversity throughout the search (enforced by MNC GA), clearly paid off in the form of better solutions in later generations.

Second, there are several issues that remain to be addressed. For example, could an equally acceptable or even a better result be obtained by some simple search strategy? Not likely, for two reasons. First, the problem is computationally complex. On a 20x21 grid there are 420 potential well positions. According to the formulation used in the paper, each pumping/recharge well configuration can contain anywhere from one to ten wells. This gives us

$$\sum_{i=1}^{10} \binom{420}{i}$$

combinations of well positions to explore. This is a huge number. Secondly, most of the classical optimization methods locate the peaks of a multi-modal function one at a time, in a sequential

manner. The merit of the MNC GA is in finding all the peaks simultaneously in one sweep. Perhaps this one sweep takes more effort than n successive sweeps made by a classical method, where n is the number of peaks. This question can be addressed by making an estimate of the computational effort of both approaches using the "big O" notation.

What about the influence of the SUTRA simulation code on the performance of MNC GA? The SUTRA code only solves the partial differential equation simulating the flow of water and the resulting solution is used only in evaluating the fitness function used in the MNC GA. It has a major influence on the time taken to get a solution, but it should have no influence on the performance of the GA itself.

Finally, more research is also needed on the GA front. What are the benefits of bigger population size and more generations? The issue here practical. To adequately address this issue, we need another way of obtaining the values for the SUTRA dependent objectives to minimize the running time. We will be looking into other ways to calculate contaminant removed and regulatory limits that although may not be as precise will give us the ranking of the solutions for each objective. Additionally we can include heuristics in the mutation operators to add, move, or delete wells that will improve the overall ranking of the individual. Finally the chromosome can be extended to include other well parameters like pumping rate to evolve solutions with the complete pumping schedule. Adding well location constraints can easily be handle in the genetic operators also.

Acknowledgements

The authors wish to acknowledge the assistance received from Virginia Johnson. She gave us copies of SUTRA code and helped us understand the intricacies of using it. Some comments made by Dr. Leah Rogers on the first draft of the manuscript were also helpful. Finally, the authors wish to thank one anonymous reviewer for identifying areas where some clarification

was warranted. Work reported here is supported in part by a grant from the Institute of Scientific Computing Research of the Lawrence Livermore National Laboratory.

References

- Cavicchio, D. J. Adaptive Search Using Simulated Evolution, Doctoral Dissertation, University of Michigan, Ann Arbor, MI, 1970.
- Cedeño, W. and V. Vemuri, Assembly of DNA restriction-fragments using genetic algorithms, *Evolutionary Computation*, 2(4): 321-345, MIT Press. 1995.
- Cedeño, W. and V. Vemuri, "Dynamic multi-modal function optimization using genetic algorithms", *Proc. of the XVIII Latin-American Informatics Conference*, Las Palmas de Gran Canaria, Spain, August 1992.
- De Jong, K. A. An analysis of the behavior of a class of genetic adaptive systems, Doctoral dissertation, University of Michigan, *Dissertation Abstracts International* 36(10), 5140B, 1975.
- De Neufville, R. *Applied Systems Analysis: Engineering Planning and Technology Management*, McGraw-Hill Publishing Co, 1990.
- Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading MA, 1989.
- Goldberg, D. E., K. Deb, and J. Horn, Massive multimodality, deception, and genetic algorithms, *Parallel Problem Solving From Nature 2*, Elsevier Science Publishers, 37-46, 1992.
- Holland, J. H. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Harbor, 1975.
- Horn, J. and N. Nafpliotis, "Multi-objective optimization using the niched Pareto genetic algorithm," *IlliGAL Report No. 93005*, Department of Computer Science, University of Illinois, Urbana-Champaign, July 1993.
- Johnson, V. M. and L. L. Rogers, "Location analysis in groundwater remediation using artificial neural networks and Monte Carlo simulations," *Preprint UCRL-JC-117289*, Lawrence Livermore National Laboratory, June 1994.
- LLNL, "Record of Decision for the Livermore National Laboratory Livermore Site, UCRL-AR-109105, July 15, 1992.
- Rogers, L. L., and F. U. Dowla, "Optimal groundwater remediation using neural networks and the genetic algorithm", *Water Resources Research*, Vol. 30, No. 2, pp 458-481, 1994.
- Rogers, L. L., F. U. Dowla and V. M. Johnson, "Optimal field-scale groundwater remediation using neural networks and genetic algorithms," *Preprint UCRL-JC-113773*, Lawrence Livermore National Laboratory, March 1994.
- Rogers, L. L., "Optimal groundwater remediation using artificial neural networks and the genetic algorithm", Ph.D. Thesis, *Preprint UCRL-LR-114125*, Lawrence Livermore National Laboratory, August 1992.
- Reid, R. W. and V. Vemuri, "On the Noninferior Index Approach to Large-Scale Multi-Criteria systems", *Journal of Franklin Institute* 291:241-254, 1971.

- Syswerda, G., Uniform crossover in genetic algorithms, Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, CA, June 1989.
- Whitley, D. GENITOR: a different genetic algorithm, Proceedings of the Rocky Mountain Conference on Artificial Intelligence, Denver Colorado, 1988.
- Winston, P. H., Artificial Intelligence, third edition, Addison-Wesley, Reading, MA, 518-528, 1992.
- Vemuri, V. and W. Cedeño. Multi-niche Crowding for Multimodal Search, in *Genetic Algorithms and Applications*, (Ed.) L. Chambers, CRC Press (in print)
- Vemuri, V. and W. J. Karplus. Digital Computer Treatment of Partial Differential Equations, Prentice Hall, Engle Wood Cliffs, 1981.
- Vemuri, V. Modeling of Complex Systems: An Introduction. Academic Press, 1978.
- Vemuri, V. "Multiple Objective Optimization in Water Resource Systems," Water Resources Research 10:44-48, 1974.
- Vemuri, V. and W. J. Karplus. Identification of Nonlinear Parameters of Ground Water Basins via Hybrid Computation. Water Resources Research 5:172-185, 1969.
- Voss, C. I. SUTRA, Saturated-Unsaturated Transport: A finite-element simulation model for saturated-unsaturated, fluid-density-dependent groundwater flow with energy transport or chemically reactive single-species solute transport, U. S. Geological Survey, Water Resources Investigations Report 84-4369, 1984.

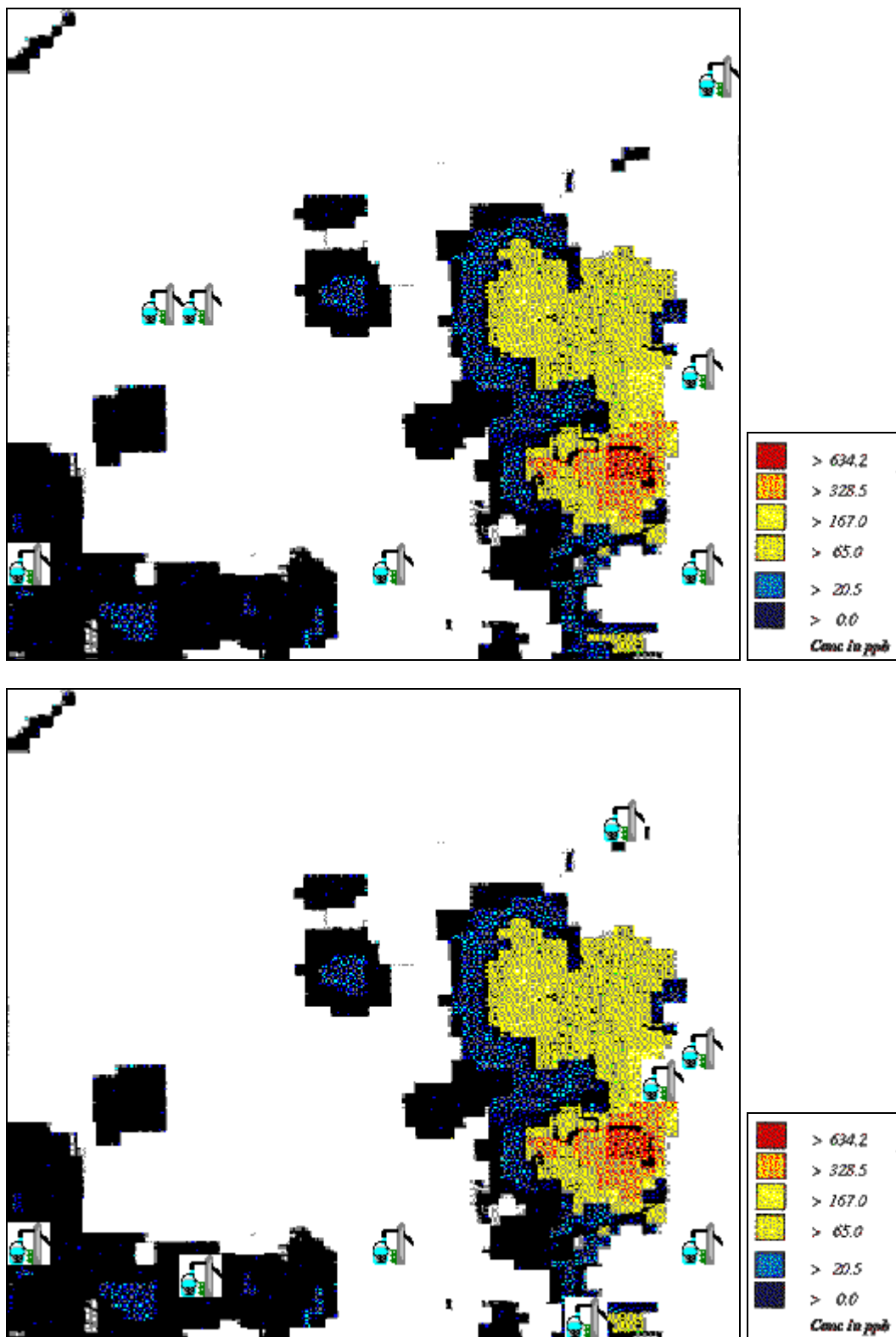


Figure 4: Well configurations for fourth (top) and fifth (bottom) solutions in Table 1.

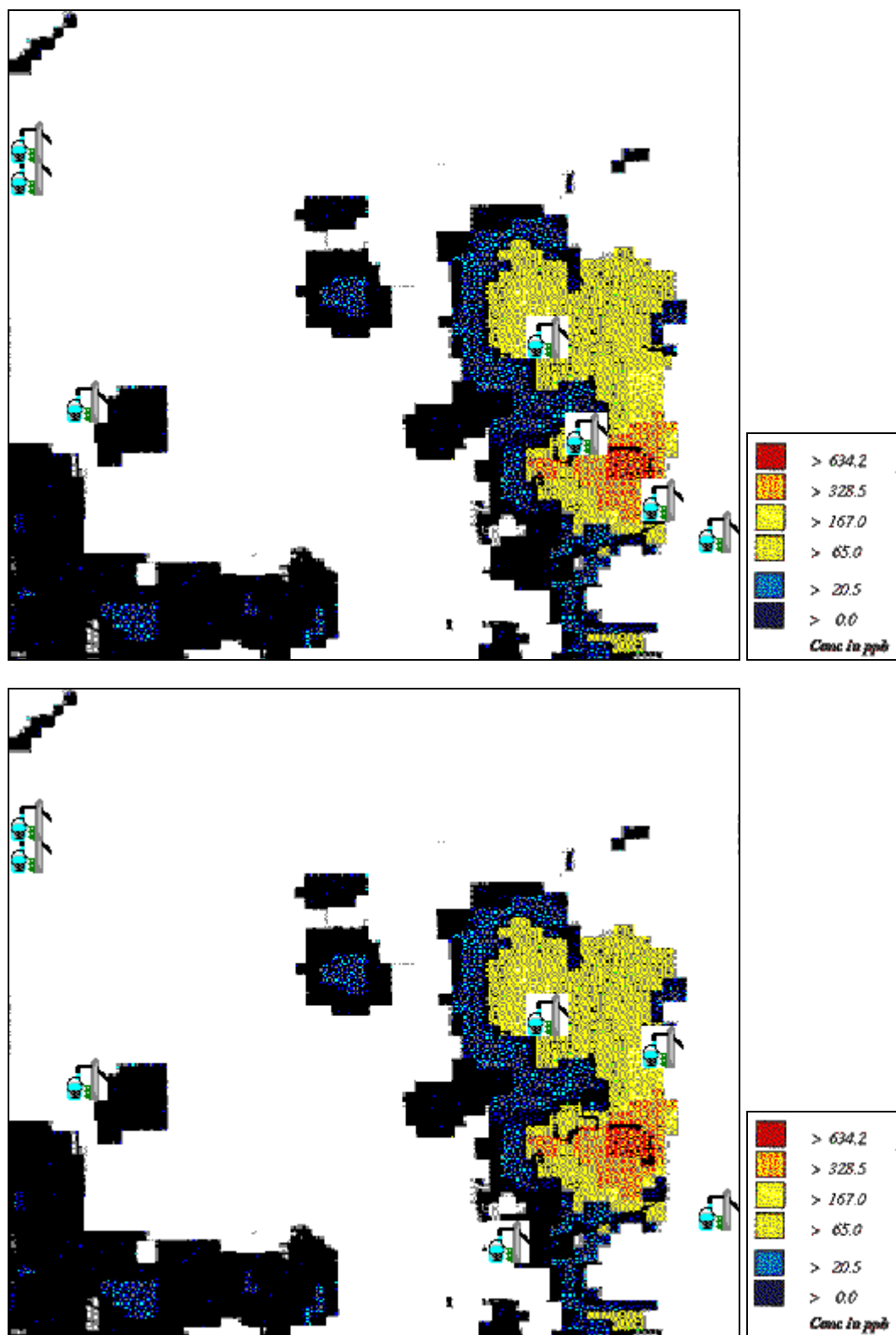


Figure 5: Well configurations for the fifth (top) and sixth (bottom) solutions in Table 2.