INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2021; 11(1):15-30

BI-OBJECTIVE OPTIMIZATION OF RESERVOIR OPERATION BY MULTI-STEP PARALLEL CELLULAR AUTOMATA

B. Oghbaei, MH. Afshar and A. Afshar[*, †]
School of Civil Engineering, Iran university of Science and technology, Tehran, Iran

ABSTRACT

Parallel Cellular Automata (PCA) previously has been employed for optimizing bi-objective reservoir operation, where one release is used to meet both objectives. However, if a single release can only be used for one objective, meaning two separate sets of releases are needed, the method is not applicable anymore. In this paper, Multi-Step Parallel Cellular Automata (MSPCA) has been developed for bi-objective optimization of single-reservoir systems' operation. To this end, a novel cellular automata formulation is proposed for such problems so that PCA's incapability when dealing with them will be overcome. In order to determine all releases throughout the operation period, in each iteration – unlike PCA – two updates take place so as to calculate releases individually. Since a bi-objective problem in Dez reservoir (in southern Iran) has been solved by PCA in earlier works, the same data is used here. The results are given for a 60-months operation period, and to evaluate this method, the results of Non-Dominated Sorting Genetic Algorithm (NSGAII) is also given for the same problem. The comparison shows MSPCA, beside remarkable reduction in computational costs, gives up solutions with higher quality as well.

Keywords: reservoir operation; multi-objective optimization; multi-step parallel cellular automata.

Received: 20 September 2020; Accepted: 10 January 2021

1. INTRODUCTION

Among the different parts of water resources systems, reservoirs play one of the most significant roles – with operation objectives of water supply, hydropower generation, flood control, sediment flushing, etc. In turn, reservoir operation optimization as well is of significance within water resources management area. In order to determine an operation plan,

[*]Corresponding author: School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran
[†]E-mail address: a_afshar@iust.ac.ir (A. Afshar)

releases and/or water storages should be decided during operation period. Yeh [1] has done a thorough study on how reservoirs should be managed.

Today, because of the potentials of water resources, various aspects are taken into consideration when dealing with them. This leads to the importance of multi-objective optimization (hereon MOO) of those resources, such as reservoirs. Features of MOO problems, along with the complexities arose from mathematical formulation of reservoir operation problems, result in non-linearity, being of large-scale, uncertainty, and so on. We need to deploy and develop methods with high efficiency in order to address these issues. Gunantara [2] has provided a review of MOO methods and its applications.

Whenever tackling MOO problems, more often than not two approaches – namely classic and evolutionary algorithms – are tried. Classic approaches simplify the solve procedure with turning a MOO problem into a multiple of single-objective optimization (hereon SOO) problems; albeit they generally get into trouble when the problem involves complexity. With evolutionary algorithms, they can solve a broader range of problems, however they too have flaws like late convergence. These are some of their application within the area of reservoir operation: Chen et al. [3], Reddy and Kumar [4], and Li et al. [5]. In a nutshell, developing methods which can arrive at proper solutions, while not having the issues of the above-mentioned approaches, is of huge importance. Deb [6] and Coello et al. [7] conducted in-depth surveys on classic approaches and evolutionary algorithms that were used in the past, and Wurbs [8] has provided a thorough study on optimization of multi-purpose reservoirs.

Cellular Automata (CA) has drawn attention in water resources area for a number of years now, as its capability has been shown. It is based on developing a graphical grid with accordance to simple rules. Within this grid – which is produced with "cells" – considering the chosen neighborhood, rules are specified to update cells' states. The most essential task when employing CA is formulating the problem regarding CA's fundamental components (cell, cell state, neighborhood, and updating rule). Although it was meant for graphical systems simulation at first, some were able to use it for optimization purposes as well.

Its application for single-objective optimization (within water resources area) goes back to 2006 when Khu and Keedwell [9] used CA to seed the initial population of genetic algorithms, which lead to significant result improving. Since then, it has been used both for generating good initial solutions, Afshar et al [10], and as a stand-alone optimizer in several researches, Guo et al. [11] and Afshar and Shahidi [12]. Recently, its application within MOO problems were extended, Afshar and Hajiabdi [13], and [14] proposed Weighted CA (WCA), and Parallel CA (PCA) respectively, although their weaknesses and strengths when tackling MOOs have not been studied thoroughly yet.

Parallel Cellular Automata has been applied to reservoir-operation problems in which the reservoirs are those that use one release for all their objectives. In other words, in a bi-objective problem for instance, the objectives are dealt with in a successive form, with one of them being a non-consumptive one. Now, the formulation for such problems cannot be used for reservoirs with, ironically, parallel objectives. In this study, PCA's weakness for this sort of problems is going to be overcome. Since in previous works, PCA has been used for Dez reservoir operation optimization, the same data is also used here, assuming the water that goes through turbines cannot be used for water supply, or in other words, hydropower generation is not non-consumptive.

2. MATERIALS AND METHODS

2.1 Multi-objective optimization

When doing a single-objective optimization, one can explicitly say which solution is the best among a set of solutions, since there is only one metric for comparing them: that single objective function; and the solution with the lowest objective function – in a minimization problem – is the optimum. If not dictated by the method (by which the optimization is being done) it is not even necessary to store solutions, as there is, at each iteration, only one best-so-far solution found. But the rules of the game will change if there are more than one objective.

Take a bi-objective optimization for example, what if a solution is better than another with respect to one of the objective functions, and is worse with respect to the other, how would the better solution be identified then? In order to compare solutions, we need to find out which one dominates the other; solution A dominates solution B, if it is not worse than solution B in terms of all objective functions, and also is strictly better than B in at least on objective, otherwise A does not dominates B. This feature of MOO eradicates the eventuality of having a single best solution, hence the existence of Pareto.

MOO problems are mostly of conflicting nature, meaning the improvement in one objective ultimately will result in deteriorating the other(s), and in turn, there is not necessarily a solution that dominates all the rest. Therefore, unlike SOO, a number of solutions should be reported as answer. Pareto front is the set of those solutions which are not dominated by any other solutions in the search space. This will dictate storing all non-dominated solutions, up to the current iteration, since they are potentially on Pareto front – unless a new solution dominates them. Note that as the size of Pareto grows, the amount of computation needed to compare solutions increases as well.

Quality of a Pareto is based on its proximity (convergence to the true Pareto front) and diversity (extent of Pareto solutions). Hypervolume is a parameter which capture these features both. In order to calculate it, a reference point should be chosen with heed to the endpoints of Pareto front. Hypervolume is the volume (or the area, in a bi-objective optimization) of the objective space dominated by a given pareto. Here, for simpler comparison, it is normalized with the rectangular area specified with reference point and ideal solution. Fig. 1. illustrates a schematic representation of hypervolume (Salazar et al. [15]).

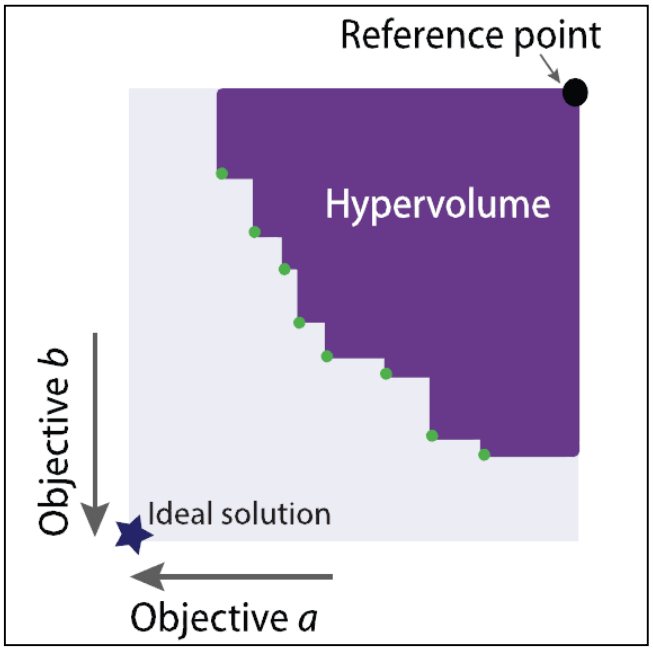


Figure 1. Hypervolume illustration

2.2 Cellular automata

2.2.1 Cellular automata history

CA was firstly proposed by Stanislaw Ulam and John von Neumann in 1940s, when they were working on different projects at Los Almos National Laboratory. Ulam was studying the growth of crystals, and how lattice network can be used for that purpose; while Neumann was engaged with self-replicating systems. Later on, they started to work together, and while trying to create a method for calculating liquid motion, they considered liquid as discrete units (Ulam's idea) whose motions were calculated according to their neighbors' behaviors, hence the very first cellular automata. After introducing CA publicly, Neumann [16] and Ulam [17], others started using it for modeling in their respective domains. Since then CA has been used for several simulation purposes in different areas.

Huge number of CA models has been suggested in order to simulate various phenomena: Alt et al [18] used CA for modelling biological systems both in intracellular activity and cluster levels of cells. Schonfisch [19], in physics, took CA to analyze dynamical systems, such as interaction of particles, and clustering of galaxies. Waksman [20] used CA to find an optimum solution to the firing squad synchronization problem. Marine et al [21] used CA for simulation of dispersion of pollutants. Karafyllidis nad Thanailakis [22] predicted forest fire with a cellular automata method. Even for drug therapies (HIV infection cases) Sentos and Coutinho [23] did some researches with CA.

2.2.2 Cellular automata components

Whenever using CA, four basic components should be determined: cells, cell state,

neighborhood, and transition rule. Cellular Automata consists of a number of cells which are represented by their states; at every modelling step, cell states are updated according to their corresponding cells' neighborhood and transition rule (also called updating rule).

Cells and cell states are selected according to the modelling purposes, whereas neighborhood of a given cell are those that their states would be shifted if the cell's is. As for the transition rule, it can be derived either analytically, or ad-hoc, or with the assistance of numerical methods; and regardless, it should be a function of current state of a given cell, and its neighbors. It is also recognized that analytically-derived updating rules lead to better results, compared to the other two (Afshar and Hajiabdi [9]).

2.2.3 Parallel cellular automata

PCA does a MOO – like evolutionary algorithms – in one act, as in there is no need to run the model several times in order to get to an approximation set of Pareto front. Basically, it is a number of SOOs (equal to the total number of objective functions) which provides us with a pareto as a result of interchanging solutions between themselves.

The idea is that, in bi-objective optimization for instance, two CA models are devised to do two single-objective optimizations individually, which normally will provide us with the endpoints of Pareto front. Now, in order to find the points in the middle we have to inject some sort of exploration into the model. To that end, we exchange the solutions of single-objective CA models on a random basis, with two mechanisms: exchange solutions with an arbitrary point on Pareto archive (PCA1), and mutual solution exchange between CA models (PCA2); Pareto archive is the set of non-dominated solutions found up to the current iteration.

To decide about the occurrence of an exchange, at each iteration a random variable is uniformly generated between 0 and 1, and if it is less than P_{it} the exchange takes place.

$$P_{it} = \lambda \times exp\left(\frac{-\left(\frac{it}{it^{max}}\right)}{1 - \frac{it}{it^{max}}}\right) \tag{1}$$

P_{it} is equal to λ at first, and as PCA goes on it decreases. This will see to it that earlier during the run solutions are exchanged with more frequency, hence the exploration; and as it comes close to the end, exploration is limited and endpoints of Pareto are exploited.

2.2.4 Multi-step parallel cellular automata (MSPCA)

When solving reservoir-operation problems with only one set of decision variables (either releases or storage volumes) to be optimized, updating rules can be rather easily derived with Newton-Raphson method. However, when there are more than one set of decision variables, independent equations are not enough anymore, which prevents us from calculating analytical updating rule.

One way to tackle this issue is to take a Sequential-Line-Search approach: assuming all sets of decision variables are constant except one, and find the analytical updating rule then. After updating the first set of decision variables, the other set can then be optimized using another rule.

In this case (Dez reservoir), we have two sets of decision variables – namely water releases for downstream demand and the water that goes through turbines for hydropower generation – so we have to formulate a 2-Step Parallel Cellular Automata (2SPCA). Assuming one set of releases is constant will able us to derive analytical transition rule (freedom degree is decreased). This would be step 1; next, with the help of mathematical programming the other set is also optimized, and we have a brand-new solution. Flowchart of the proposed method can be drawn as follows:

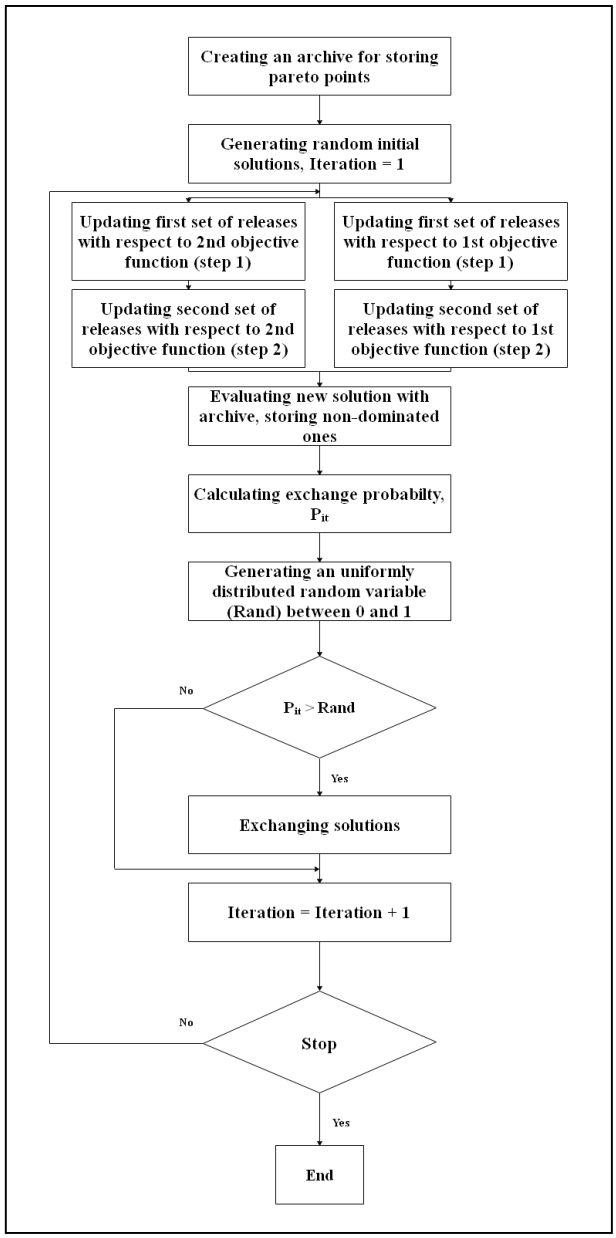


Figure 2. A 2-Step Parallel Cellular Automata flowchart for reservoir operation optimization

With 2SPCA, as it is a combination of a number of single-objective optimizations, primarily it goes for two endpoints of Pareto; and even though exchanging solutions steps up exploration aspect of the method, it is still required to somehow get 2SPCA to not skip the area in which the middle points of Pareto front are. One way to settle this, is reducing updating terms. In this study, factor α is used to do so by multiplying updating terms at step 2 by a number between 0 and 1 (α). Fig. 3 shows a pareto, found by 2SPCA, with not-reduced updating terms i.e. $\alpha = 1$.

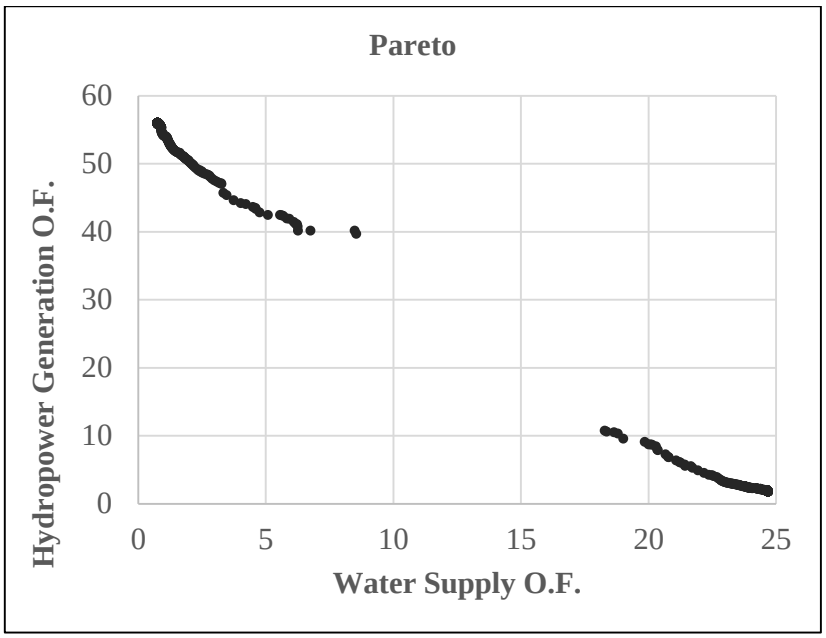


Figure 3. An example of 2SPCA's final pareto with $\alpha = 1$

2.3 Dez reservoir

In previous studies, PCA has been employed for optimizing bi-objective operation of Dez reservoir when one single release would be used for both hydropower generation, and downstream water demand; whereas when they are separately operated – as in the water, that goes through turbines, cannot be used for downstream water demand and vice versa – both mathematical and cellular automata formulation of the problem change to the point that if it is not taken into consideration, the model will not end up at an even remotely good approximation set of Pareto front.

Dez reservoir (in southern Iran) has a total volume of 2510 MCM, which is an outcome of maximum and minimum allowable storage volumes of, respectively, 3340 and 830 MCM. The storage volume of the reservoir at the very beginning of operation period is taken to be 1430 MCM. Releases cannot be less than 0, and violate a maximum boundary of 1000 MCM.

The hydroelectric plant consists of eight 80.8 MW units, with a total capacity of 650 MW, and 90 percent efficiency. As they work 10 hours per day, the plant factor equals 0.417. Tail water level (TWL) of the hydroelectric plant is considered to be constant at 172 meters.

2.3.1 Mathematical formulation

In Dez reservoir, the objective of reservoir operation is to find a set of releases in a way that downstream demand is met as much as possible (water supply) or that energy is generated as close to total capacity of hydroelectric plants as possible (hydropower generation). The corresponding objectives are defined, respectively, as below:

$$Min\ F_{ws} = \sum_{t=1}^{NT} \left(\frac{D_t - RW_t}{D_{max}}\right)^2 \tag{2}$$

$$Min\ F_{hy} = \sum_{t=1}^{NT} \left(\frac{P^{max} - P_t}{P^{max}}\right)^2 \tag{3}$$

NT is the number of time periods, D_t is water demand at period t (MCM), RW_t is water release from reservoir for downstream demand at period t (MCM), D_{max} is the maximum demand at all periods (MCM), P^{max} is total capacity of hydroelectric plant (MW), and P_t is the power generated by hydroelectric plant at period t (MW). Also P_t is calculated by the following formula:

$$P_t = min\left(\frac{\rho \times g \times \eta \times RH_t \times H_t}{10^6 \times PF \times C}\ ,\ P^{max}\right) \tag{4}$$

where ρ is water density, g is the gravity acceleration, η is the hydropower plant efficiency, RH_t is the water that goes through turbines at period t (MCM), PF is the plant factor, C is a coefficient which converts the release volume (MCM) to discharge (m^3/s), and H_t is the effective head (m) of the hydropower plant at period t which is defined as:

$$H_t = \left(\frac{E_t + E_{t+1}}{2}\right) - TWL \tag{5}$$

here TWL is the tail water level of hydropower plant, and E_t is the water level elevation (m) of reservoir at period t which can be calculated using this equation:

$$E_t = aS_t^3 + bS_t^2 + cS_t^2 + d \tag{6}$$

It demonstrates the volume-elevation curve of the reservoir which is obtained by fitting the equation to the observed data. The coefficients of volume-elevation given by equation (5) are:

$$a = 1.526 \times 10^{-9},\ b = -1.37 \times 10^{-5},\ c = 0.058720,\ d = 249.83364 \tag{7}$$

The operation is also subjected to the following constraints:

$$S_{t+1} = S_t + I_t - RW_t - RH_t\ ,\ t = 1, 2, \ldots, NT \tag{8}$$

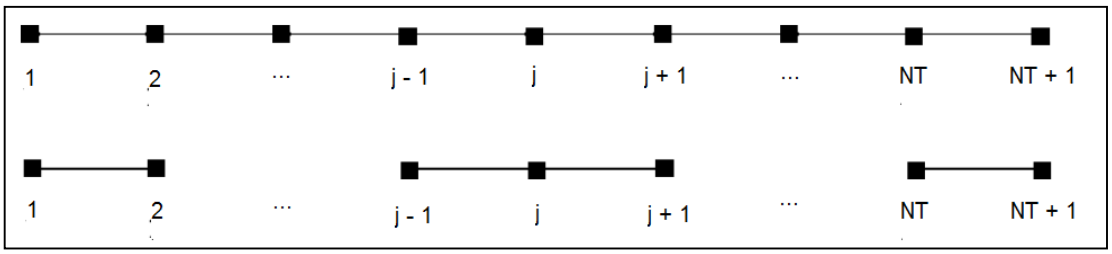
$$S^{min} \le S_t \le S^{max}\ ,\ t=1,\ 2,\ldots,\ NT \tag{9}$$
$$R^{min} \le RW_t \le R^{max}\ ,\ t=1,\ 2,\ldots,\ NT \tag{10}$$
$$R^{min} \le RH_t \le R^{max}\ ,\ t=1,\ 2,\ldots,\ NT \tag{11}$$

S_t is the water storage of the reservoir at period t (MCM), I_t is the water inflow into the reservoir at period t (MCM), S^{min} and S^{max} are, respectively, minimum and maximum value of water storage (MCM), and lastly R^{min} and R^{max} are, again respectively, minimum and maximum possible values of release from the reservoir (MCM).

2.3.2 Cellular automata formulation

In order to apply CA to optimization problems the four above-mentioned components have to be defined beforehand. Depending on what is chosen as cell state, the other three components will vary as well.

Here we take water storage (*S*) as cell state, and cells would be the beginning of each period which actually corresponds to water storage of the reservoir; this leads to a cellular automata with $NT + 1$ cells. As for the neighborhood, one cell from each side of a given cell is considered to be the neighbor of that cell, except for boundary ones ($t = 1$ and t = $NT + 1$) which follow the same regulation, but since there is just one cell next to them, their neighborhood would be including only that one cell. Fig. 4 illustrates a schematic representation of the cells.



(a) (b) (c)

Figure 4. schematic representation of cells and their neighbors: (a) left boundary cell, (b) middle cell *j*, and (c) right boundary cell

Defining the first three components, now is the time to do the same for transition rule. Using Newton-Raphson method and each objective function, the transition rule for cell *j* at step 1 can be obtained like this:

$$\left(\Delta S_j^{ws}\right)^k = + \frac{\left(\frac{\partial RW_{j-1}}{\partial S_j}\right)\left(D_{j-1}\text{-}RW_{j-1}^k\right) + \left(\frac{\partial RW_j}{\partial S_j}\right)\left(D_j\text{-}RW_j^k\right)}{\left(\frac{\partial RW_{j-1}}{\partial S_j}\right)^2 + \left(\frac{\partial RW_j}{\partial S_j}\right)^2} \tag{12}$$

$$\left(\Delta S_j^{hy}\right)^k = + \frac{\left(\frac{\partial P_{j-1}}{\partial S_j}\right)\left(P^{max}\text{-}P_{j-1}^k\right) + \left(\frac{\partial P_j}{\partial S_j}\right)\left(P^{max}\text{-}P_j^k\right)}{\left(\frac{\partial P_{j-1}}{\partial S_j}\right)^2 + \left(\frac{\partial P_j}{\partial S_j}\right)^2} \tag{13}$$

In order to satisfy problem's constraints, updating terms are box-constrained:

$$S_j^{min} - S_j^k \le \left(\Delta S_j^{ws}\right)^k \le S_j^{max} - S_j^k \tag{14}$$

$$R^{min} - RW_j^k \le \left(\Delta S_j^{ws}\right)^k \le R^{max} - RW_j^k \tag{15}$$

$$RW_{j-1}^k - R^{max} \le \left(\Delta S_j^{ws}\right)^k \le RW_j^k - R^{min} \tag{16}$$

$$S_j^{min} - S_j^k \le \left(\Delta S_j^{hy}\right)^k \le S_j^{max} - S_j^k \tag{17}$$

$$R^{min} - RH_j^k \le \left(\Delta S_j^{hy}\right)^k \le R^{max} - RH_j^k \tag{18}$$

$$RH_{j-1}^k - R^{max} \le \left(\Delta S_j^{hy}\right)^k \le RH_j^k - R^{min} \tag{19}$$

At step 2, updating rules are of ad-hoc nature, and using mathematical programming the amount by which cell states are to be updated will be calculated. Cells, cell states, and neighborhood are as specified before.

- water supply single-objective optimization:

$$\begin{gathered}
\Delta S_j^{ws} = (X_1)\left(D_j\text{-}RW_j\right) - (1\text{-}X_1)\left(D_{j-1}\text{-}RW_{j-1}\right) \\
\Delta RW_j^{ws} + \Delta RH_j^{ws} = \Delta S_j^{ws} \\
(X_1)\left(\Delta RW_j^{ws}\right) + (1\text{-}X_1)\left(\Delta RH_j^{ws}\right) = \Delta S_j^{ws} \\
\Delta RW_{j-1}^{ws} + \Delta RH_{j-1}^{ws} = \Delta S_j^{ws} \\
(X_2)\left(\Delta RW_{j-1}^{ws}\right) + (1\text{-}X_2)\left(\Delta RH_{j-1}^{ws}\right) = -\Delta S_j^{ws} \\
X_1 = \begin{cases} 1\ ; & RW_j \ne D_j \\ 0\ ; & RW_j = D_j \end{cases} \\
X_2 = \begin{cases} 1\ ; & sign\left(D_j\text{-}RW_j\right) \times sign\left(D_{j-1}\text{-}RW_{j-1}\right) = -1 \\ 1\ ; & D_j - RW_j = 0 \\ 0\ ; & otherwise \end{cases}
\end{gathered} \tag{20}$$

In order to satisfy problem's constraints, updating terms are box-constrained with the following. Note that, M represents a big number in such a way that if its coefficient equals 1, the corresponding constraint is never activated.

$$R^{min}\text{-}RW_j^k - (1\text{-}X_1)(M) \le \left(\Delta S_j^{ws}\right)^k \le R^{max}\text{-}RW_j^k + (1\text{-}X_1)(M) \tag{21}$$

$$R^{min}\text{-}RH_j^k - (X_1)(M) \le \left(\Delta S_j^{ws}\right)^k \le R^{max}\text{-}RH_j^k + (X_1)(M) \tag{22}$$

$$RW_{j-1}^{k} - R^{max} - (1-X_2)(M) \le \left(\Delta S_j^{ws}\right)^k \le RW_{j-1}^{k} - R^{min} + (1-X_2)(M) \quad (23)$$

$$RH_{j-1}^{k} - R^{max} - (X_2)(M) \le \left(\Delta S_j^{ws}\right)^k \le RH_{j-1}^{k} - R^{min} + (X_2)(M) \quad (24)$$

- hydropower generation single-objective optimization:

Here, we use an artificial variable *DH* which is equal to the amount of water that had gone through turbines, it would have resulted in generating P^{max} at period *t*:

$$DH_t = \frac{10^6 \times PF \times C}{\rho \times g \times \eta \times H_t} \times P^{max} \quad (25)$$

Knowing the value of *DH* at each period, we can now calculate updating terms:

$$\Delta S_j^{hy} = (X_1)\left(DH_j - RH_j\right) - (1-X_1)\left(DH_{j-1} - RH_{j-1}\right)$$

$$\begin{gathered}
\Delta RW_j^{hy} + \Delta RH_j^{hy} = \Delta S_j^{hy} \\
(1-X_1)\left(\Delta RW_j^{hy}\right) + (X_1)\left(\Delta RH_j^{hy}\right) = \Delta S_j^{hy} \\
\Delta RW_{j-1}^{hy} + \Delta RH_{j-1}^{hy} = \Delta S_j^{hy} \\
(1-X_2)\left(\Delta RW_{j-1}^{hy}\right) + (X_2)\left(\Delta RH_{j-1}^{hy}\right) = -\Delta S_j^{hy} \\
X_1 = \begin{cases} 1 ; & RH_j \neq DH_j \\ 0 ; & RH_j = DH_j \end{cases} \\
X_2 = \begin{cases} 1 ; & sign\left(DH_j - RH_j\right) \times sign\left(DH_{j-1} - RH_{j-1}\right) = -1 \\ 1 ; & DH_j - RH_j = 0 \\ 0 ; & otherwise \end{cases}
\end{gathered} \quad (26)$$

In order to satisfy problem's constraints, updating terms are box-constrained with the following. Note that, M represents a big number in such a way that if its coefficient equals 1, the corresponding constraint is never activated.

$$R^{min} - RW_j^k - (X_1)(M) \le \left(\Delta S_j^{hy}\right)^k \le R^{max} - RW_j^k + (X_1)(M) \quad (27)$$

$$R^{min} - RH_j^k - (1-X_1)(M) \le \left(\Delta S_j^{hy}\right)^k \le R^{max} - RH_j^k + (1-X_1)(M) \quad (28)$$

$$RW_{j-1}^{k} - R^{max} - (X_2)(M) \le \left(\Delta S_j^{hy}\right)^k \le RW_{j-1}^{k} - R^{min} + (X_2)(M) \quad (29)$$

$$RH_{j-1}^{k} - R^{max} - (1-X_2)(M) \le \left(\Delta S_j^{hy}\right)^k \le RH_{j-1}^{k} - R^{min} + (1-X_2)(M) \quad (30)$$

3. RESULTS AND DISCUSSION

3.1 Parameters

Firstly, a bunch of runs are required to be made to determine the best value of λ for this problem.

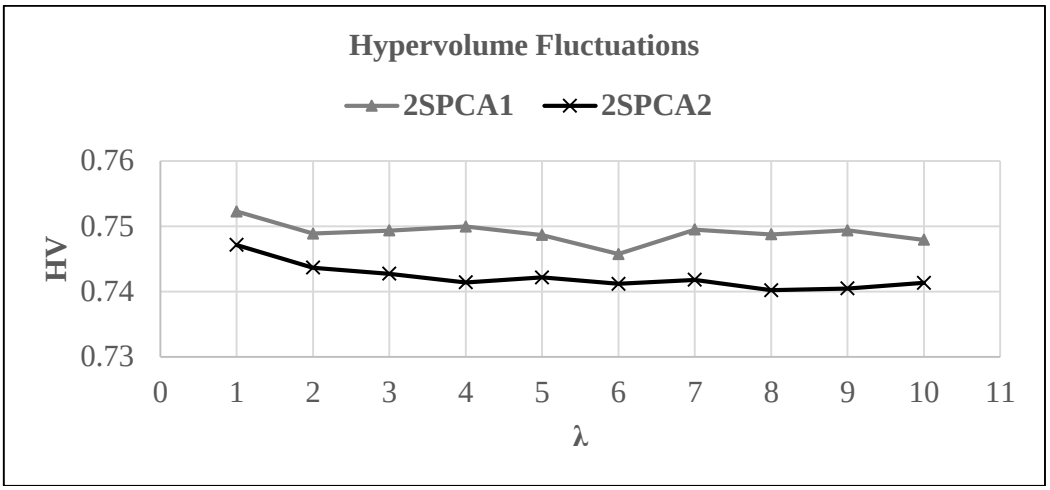


Figure 5. Hypervolumes with different initial exchange probabilities

As it can be seen, the best value for λ is 0.1 – no matter what is the exchange mechanism (2SPCA1 or 2SPCA2). Also, with decreasing trend of the graphs in mind, it is implied that exchanging solutions with higher probability (i.e. for too many times) is not desirable.

It should be highlighted there is no connection between α and either λ or the exchange mechanism for that matter, since α only affects the behavior of single-objective CA models. On one hand, too high values of it will result in not exploring a massive area of search space, and on the other, if it is too low then the method will be stopped in the wake of premature convergence. In this study, we have used a value of 0.1 for α since it avoids both of hitches mentioned.

Lastly, to be able to measure hypervolume we need to define a reference point. Here, the corresponding point on objective functions space to no releases at all is taken for that, which has the coordinates of (26.1155, 60).

3.2 Model results

Dez reservoir-operation problem for a period of 5 years (60 months) has been solved using 2SPCA; and for the purpose of evaluating results, NSGAII has been used to solve the same problem as well. Table 1 shows each methods' results. Note that first two rows are all averages for 10 runs.

Table 1: 2SPCAs and NSGAII results

Method	λ	Iterations	Time (s)	Archive/ Population size	Hypervolume	Minimum value of water supply O.F.	Minimum value of hydropower generation O.F.
2SPCA1	0.1	4829	11.14	3394	0.757	0.65	1.85
2SPCA2	0.1	4241	7.02	1843	0.752	0.67	1.86
		8765	1097	50	0.702	1.79	7.41
NSGAII	×	12478	5009	100	0.727	1.23	5.13
		19754	29844	200	0.743	0.97	4.52

2SPCA1 and 2SPCA2 give up similar values for hypervolume, although 2SPCA1 is doing better. The real difference is in archive size, where 2SPCA1 has found a lot of more non-dominated solutions, which is actually the reason why it has required more time. As for the endpoints, 2SPCA1 is doing – again – better, although it might not be of significance.

NSGAII, apart from its extremely high computational cost, could not find endpoints well enough. This has led to having lower values of hypervolume compared to 2SPCA. In addition, although increasing population size have resulted in increasing hypervolume too, the amount of time needed for it is too much to be anything close to efficient; NSGAII (with population size of 50) alone needs about 100 times the time 2SPCA1 needs, and it grows even more if population size is increased. Figs. 6 and 7 show paretos of 2SPCA1, 2SPCA2, and NSGAII (with population size of 200).

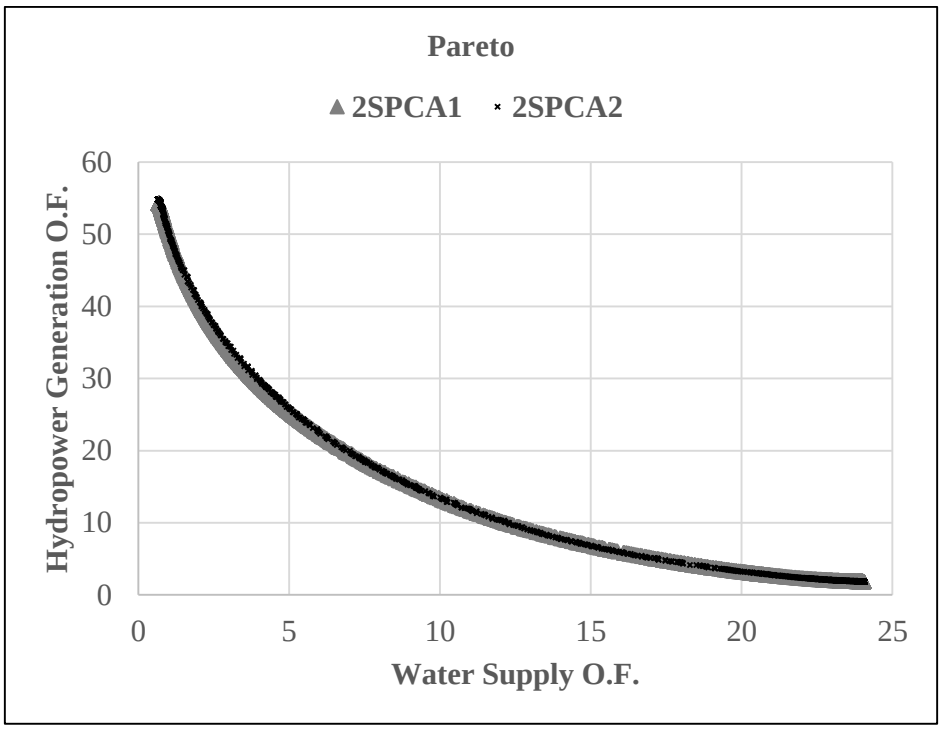


Figure 6. 2SPCA1 and 2SPCA2 paretos

The two paretos are extremely close to each other. The reason for the difference between hypervolumes is: one, 2SPCA1 has found way more non-dominated solutions which leads to dominating more amount of search space, and two, 2SPCA1 – even though slightly – has also found better solutions at endpoints which again leads to dominating more amount of search space, hence the higher hypervolume.

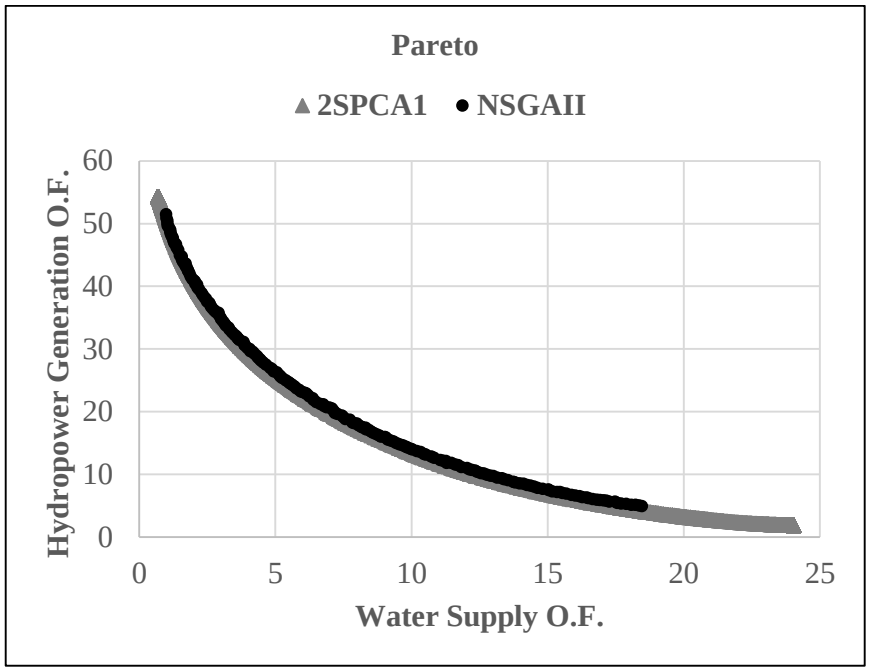


Figure 7. 2SPCA1 and NSGAII (population size of 200) paretos

NSGAII (here with a population size of 200) has done well regarding reaching a good approximation of parent front; however, it has failed to find solutions near endpoints. These issue can be addressed with more population size, but it will cost us a huge amount of time for it to converge, which is not efficient.

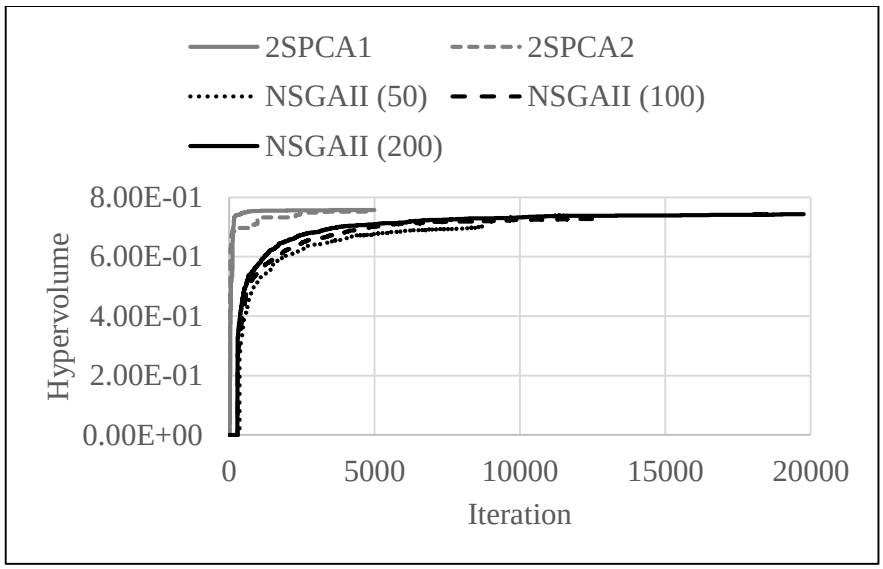


Figure 8. Hypervolume value versus iteration number

Hypervolume increase throughout the run, for each method, is shown in Fig. 8. At first, its value equals zero, regardless of the method; this is because initial solutions are generated randomly, and they are outside of the reference area – which means no proportion of reference area is dominated. 2SPCA reaches reference area much faster than NSGAII; the reason for that is: 2SPCA works with two solutions at a time and moves toward Pareto specifically, since it has the advantage of having an analytical rule, after all it is a gradient-based method. As for NSGAII, it just explores the search space, and there is no information whatsoever as to in which direction the exploration should move, resulting in a huge amount of time for reaching reference area and eventually Pareto. Also, the parenthesis in the chart legend is population size, and the more population size of NSGAII, the bigger values of hypervolume; but again, it comes with a computational cost, and yet NSGAII's hypervolume does not equal that of 2SPCA.

4. CONCLUSION

CA's capabilities in solving SOO problems have been shown in the past, and recently a new alteration of it, called Parallel Cellular Automata, was developed for MOO ones. PCA has shown efficiency over evolutionary algorithms in terms of quality of the results, and computational cost. However, it has not been applied to a wide range of problems. In this study, Multi-Step Parallel Cellular Automata (MSPCA) was proposed in order to deploy it for solving MOO problems with different characteristics from the ones that PCA was previously used to solve. This method was as well able to produce better results with comparison to Non-Dominated Sorting Genetic Algorithm. Considering hypervolume for evaluating the approximation set of global pareto front, MSPCA dominated a larger area of objective function space, compared to that of NSGAII, in addition of a huge reduction in run time. Also, MSPCA's performance can be enhanced using various exchange mechanisms.

REFERENCES

1. Yeh WWG. Reservoir management and operations models: a state-of-the-art review, *Water Resources Research* 1985; **21**(12): 1797-1818.
2. Gunantara N. A review of multi-objective optimization: methods and its applications, *Cogent Eng* 2018; **5**(1): 1502242.
3. Chen L, McPhee J, Yeh WWG. A diversified multi objective GA for optimizing reservoir rule curves. Adv Water Resour, 2007; **30**(5): 1082-1093.
4. Reddy MJ, Kumar ND. Multi-objective particle swarm optimization for generating optimal tradeoffs in reservoir operation, *Hydrol Process* 2007; **21**(21): 2897-2909.
5. Li Y, Zhou J, Zhang Y, Qin H, Liu L. Novel multi objective shuffled frog leaping algorithm with application to reservoir flood control operation, *J Water Resour Plan Manag* 2010; **136**(2): 217-26.
6. Deb K. Multi-objective optimization using evolutionary algorithms. Wiley, Chichester, UK, 2001.

7. Coello C, Lamont GB, Van Veldhuizen DA. Evolutionary algorithms for solving multi-objective problems, Springer, 5th edition, New York, USA, 2007.
8. Wurbs RA. Optimization of Multiple-Purpose Reservoir Systems Operations: A Review of Modeling and Analysis Approaches, Hydrologic Engineering Center Davis CA, 1991.
9. Keedwell E, Khu ST. Novel cellular automata approach to optimal water distribution network design, *J Comput Civ Eng* 2006; **20**(1): 49-56.
10. Afshar MH, Azizipour M, Oghbaeea B, Kim JH. *Exploring the Efficiency of Harmony Search Algorithm for Hydropower Operation of Multi-reservoir Systems: A Hybrid Cellular Automat-Harmony Search Approach*, Advances in Intelligent Systems and Computing, Bilbao, Spain, pp. 252-260.
11. Guo Y, Walters GA, Khu ST, Keedwell E. A novel cellular automata based approach to optimal storm sewer design, *Eng Optim* 2007; **39**(3): 345-64.
12. Afshar MH, Shahidi M. Optimal solution of large-scale reservoir operation problems: cellular automata versus heuristic search methods, *Eng Optim* 2009; **41**(3): 275-93.
13. Afshar MH, Hajiabadi R. A Novel Parallel Cellular Automata Algorithm for Multi-Objective Reservoir Operation Optimization, *Water Resour Manag* 2018; **32**: 785-803.
14. Afshar MH, Hajiabadi R. Multi-objective optimisation using cellular automata: application to multi-purpose reservoir operation, *Civil Eng Environ Syst* 2019; **36**(2-4): 115-32.
15. Salazar JZ, Reed PM, Quinn JD, Giuliani M, Castelletti A. Balancing Exploration, Uncertainty and Computational Demands in Many Objective Reservoir Optimization, *Adv Water Res* 2017; **109**: 196-210.
16. Von Neumann J. Theory of self-reproducing automata, Champaign, IL, University of Illinois Press, 1966.
17. Ulam S. A collection of mathematical problems. Interscience publishers, New York, 1960.
18. Alt W, Deutsch A, Dunn G. Dynamics of cell and tissue motion. Birkhauser, Basel, 1997.
19. Schonfisch B. Propagation of fronts in cellular automata, *Physic D* 2002; **80**: 433-50.
20. Waksman A. An optimum solution to the firing squad synchronization problem, *Inf Control* 1966; **9**; 67-78.
21. Marin M, Rojas-Molina V, Lopez-Cajun CS, Herrera A, Castano VM. Cellular automata simulation of dispersion of pollutants, *J Computat Mater* 2000; **18**: 132-40.
22. Karafyllidis I, Thanailakis A. A model for predicting forest fire spreading using cellular automata, *Ecol Model* 1997; **99**(1): 87-97.
23. Sentos, RMZD, Coutinho S. Dynamics of HIV approach: a cellular automata approach, *Physi Rev Letter* 2001; **87**(16): 102-4.