# Coronavirus Herd Immunity Optimizer with Tent and Circle Chaotic Maps for the *p*-Center Facility Location Problem: The Case of Davao City, Digos City and Tagum City, Philippines

Hanna Mae Lina Limpag
University of the Philippines Mindanao
Davao City, Philippines
hllimpag@up.edu.ph

Ritchie Mae T. Gamot
University of the Philippines Mindanao
Davao City, Philippines
rtgamot@up.edu.ph

## ABSTRACT

The Coronavirus Herd Immunity Optimizer (CHIO) is a relatively new nature-inspired algorithm that is inspired by the concept of herd immunity. The herd or population is said to be gradually affected by three different cases: the infected, susceptible, and immune cases. Improvements in the status of everyone in the herd is influenced by the algorithmic parameter, called the reproduction rate, or the rate at which a virus spreads. Performance of CHIO using benchmark optimization functions and comparison to well-known swarm-based algorithms have shown a good balance between exploration and exploitation. As CHIO is still in its infancy, this study attempted to use the original algorithm and its modified version to solve an NP-Hard problem, called the *p-center* problem. The modification was a simple addition of chaotic maps, tent and circle chaotic maps, to the original CHIO algorithm. Experimental and statistical results using household data sets from the Davao Region (Davao City, Tagum City and Digos City) showed that both CHIO and CHIO with chaotic maps can minimize the *p-center* problem with the former showing better results in most cases. Reproduction rate values were also varied, and results revealed that a higher basic reproductive of (BRr = 0.90) performed better than lower basic reproductive rate (BRr = 0.01).

## KEYWORDS

Metaheuristics, Nature-inspired optimization, Facility location problem, *p-center* problem

*Article Title Footnote needs to be captured as Title Note
†Author Footnote to be captured as Author Note

*PCSC2024, May 2024, Laguna, Philippines*

https://doi.org/10.1145/1234567890

**ACM Reference format:**

FirstName Surname, FirstName Surname and FirstName Surname. 2024. Coronavirus Herd Immunity Optimizer with Chaotic Maps for the *p-center* Facility Location Problem: The case of Davao City, Digos City and Tagum City, Philippines. In *Proceedings of Philippine Computing Science Congress (PCSC2024). Laguna, Philippines, 8 pages.*

## 1 Introduction

Facility location decisions show a prominent role in strategic planning of many firms, companies, and governmental organizations [1] as it involves the optimal placement of services such as education, health, commerce and many more. Such type of decision fall under the *p-center* minimax location-allocation problem.

The *p-center* problem is the location of facilities where the maximum distance between the nearest facility and its demand point is minimized [2]. Despite its simplicity and even in small-scale, the *p-center* problem is considered an NP-Hard problem [3].

The *p-center* in Euclidean space is in Equation 1:

$$\min_{(x_i, y_i)} \max_{i} \min_{j} [(a_i - x_i)^2 + (b_i - y_i)^2] \quad (1)$$

where $(a_i, b_i)$ are the coordinates of the *i*th demand point and $(x_i, y_i)$ are the coordinates of the *i*th facility. Deterministic and metaheuristic methods have been used in the past to solve the *p-center* problem.

The *p-center* problem was solved by Gaar and Sinnl [4] on a new integer programming formulation by means of branch-and-cut where cuts for customer demands points are iteratively generated. In the same study, they generated a way to use lower bound information to obtain stronger cuts which are at par with best known in literature. Computational studies with up to more than 700,000 customers and locations showed that, for many instances, the algorithm was competitive. Gaar and Sinnl [4] suggested hybridization techniques to alleviate time consuming calculations for some large-scale instances. Contardo, et al [5], on the other hand, solved the vertex *p-center* problem by proposing a row generation algorithm that iteratively solves smaller subproblems using subsets of the clients. Variable number of points are added in this subset at the end of each iteration. Computational experiments on up to 1 million clients and centers showed its scalability. Small values of *p* were solved to optimality whilst instances greater than $10^4$ were solved faster than expected. Nematian and Sadati [6] introduced a *p-center* problem with demands treated as fuzzy random variables. The problem was converted to a deterministic integer programming problem to be solved by new methods based on implementation of possibility theory and fuzzy random chance-constrained programming. Using the location of bicycle stations in Tabriz City, Iran, their method showed robustness and applicability to real cases with uncertainty.

Anonymous [7] solved the Euclidean *p-center* problem using a nature-inspired optimization algorithm called the Stud Krill Herd Algorithm (SKHA). It was applied, for the first time, to the allocation of early warning devices (EWD) in Digos City, Philippines. Algorithmic parameters were tested and showed the capacity of SKH to find possible best locations of EWDs for p = 5, 6 and 7. Yin et al [8] used a greedy randomized solution and tabu search technique to solve the vertex *p-center* problem. The resulting solution is combined with one of the elite solutions by path-relinking, which consists in exploring trajectories that connect high-quality solutions. Their method showed competitive edge over state-of-the-art algorithms in literature both in solution quality and computational efficiency. Anonymous et al [9] also used the Moth-Flame Optimization Algorithm (MFOA) and Whale Optimization Algorithm (WOA), for the first time, on the Euclidean *p-center* problem. Experimental results using the Digos City, Philippines data alongside tests on algorithmic parameters showed the ability of both algorithms to find better locations for *p*=5, 6 and 7 as iterations progress. Between the two, MFOA generally showed better solution quality over WOA.

In this study, a relatively new nature-inspired optimization algorithm called the Community Herd Immunity Optimizer (CHIO), attempts to solve the *p-center* problem, for the first time, using household data locations of Digos City, Davao City and Tagum City, Philippines. Furthermore, chaotic maps were introduced to determine how the algorithm behaves from the traditional randomized generation initial solution generation and fatality cases. This study aims to contribute to alternative approaches to the Euclidean *p-center* problem.

## 2 Community Herd Immunity Optimizer

Herd immunity is the indirect protection from an infectious disease that occurs when a population has established immunity to disease through vaccination or previous infection. CHIO is a relatively new nature-inspired optimization algorithm inspired by herd immunity developed by Al-Betar et al [10]. This algorithm aims to provide solution, $x = (x_1, x_2, \dots, x_n)$, to the minimization/maximization problem $f(x)$, where $n$ is the total number of genes per individual.

The seminal paper on CHIO compared the algorithm to seven well-known swarm-based algorithms [10]. Using 23 benchmarks functions. CHIO was able to obtain 16 out of 23 new results compared to the other methods. It was also used to solve real-world engineering optimization problems and compared with nine other set of methods. CHIO was also proven to be generally competitive in these kinds of problems. It is important to note that CHIO had a variation called Multi-Objective CHIO, or MOCHIO [11], to optimize the design of a brushless direct current motor in the domain of magnetics. Results show that MOCHIO appears to be viable and dominant.

The implementation procedure algorithm:

1. Initialize algorithmic and control parameters to include the number of initial infected cases $(C_0)$, maximum number of iterations $(\max\_iter)$, population size or total number of cases/individuals $(HIS)$ and number of genes per individual, $(n)$, basic reproduction rate $(BR_r)$ and maximum age of infected cases $(\max\_age)$.
2. Generate the herd immunity population called $HIP$ (Equation 2) of size $HIP$ with dimension $n$. Each element in the matrix is generated randomly within the range of the lower bound and upper bound of the decision variables. Additionally, a one-dimensional status vector, $S$, of size $HIS$ is initiliazed to 0. The $S$ vector later will contain the values 0 (susceptible case), 1 (infected case) or 2 (immuned case). Another one-dimensional vector, $A$, of size $HIS$, called the age vector is also initialized to zero.

$$HIP = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{HIS} & x_2^{HIS} & \cdots & x_n^{HIS} \end{bmatrix} \quad (2)$$

3. Evolve the population based on Equation 3. Depending on a generated random number, $r$, and the $BRr$, each individual will either be updated using Equations 4, 5, or 6. The indices $c$, $m$ and $v$ are randomly chosen cases from the infected, susceptible and immune case, respectively.

$$x_i^j(t+1) = \begin{cases} C(x_i^j(t)) & r \in [0, {}^1/_3\, BRr) \\ N(x_i^j(t)) & r \in [{}^1/_3\, BRr, {}^2/_3\, BRr) \\ R(x_i^j(t)) & r \in [{}^2/_3\, BRr, BRr) \\ x_i^j(t) & r \in [BRr, 1) \end{cases} \quad (3)$$

$$C(x_i^j(t)) = x_i^j(t) + \mathrm{r} \times \left(x_i^j(t) - x_i^c(t)\right) \quad (4)$$

$$N(x_i^j(t)) = x_i^j(t) + \mathrm{r} \times \left(x_i^j(t) - x_i^m(t)\right) \quad (5)$$

$$R(x_i^j(t)) = x_i^j(t) + \mathrm{r} \times \left(x_i^j(t) - x_i^v(t)\right) \quad (6)$$

4. Update the status and age vectors using Equation 7. The current solution will only be replaced by a new better solution (better immunity rate) else the age vector for this solution/case is incremented by 1.

$$(S_{j,} A_j) = \begin{cases} (1,1) & if\ Equation\ 8 \\ (2,0) & if\ Equation\ 9 \end{cases} \quad (7)$$

$$f\left(x^j(t+1)\right) < \frac{f\left(x^j(t+1)\right)}{\Delta f(x)} and\ S_j = 0\ and\ is\_Corona(x^j(t+1)) \quad (8)$$

$$f\left(x^j(t+1)\right) > \frac{f\left(x^j(t+1)\right)}{\Delta f(x)}\ and\ S_j = 1 \quad (9)$$

$$is\_Corona(x^j(t+1)) = \begin{cases} 1 & If\ new\ case\ is\ inherited\ from\ infected\ case \\ 0 & otherwise \end{cases} \quad (10)$$

5. Update fatality cases by regenerating a new random case if $A_j > max\_age$ and $S_j = 1$. The status and age values for this case is set to 0 and 1, respectively.
6. The algorithm iterates steps 3 to 5 until $\max\_iter$ iterations are reached.

For a more helpful understanding, these six primary steps are also represented as a flowchart using Figure 1.

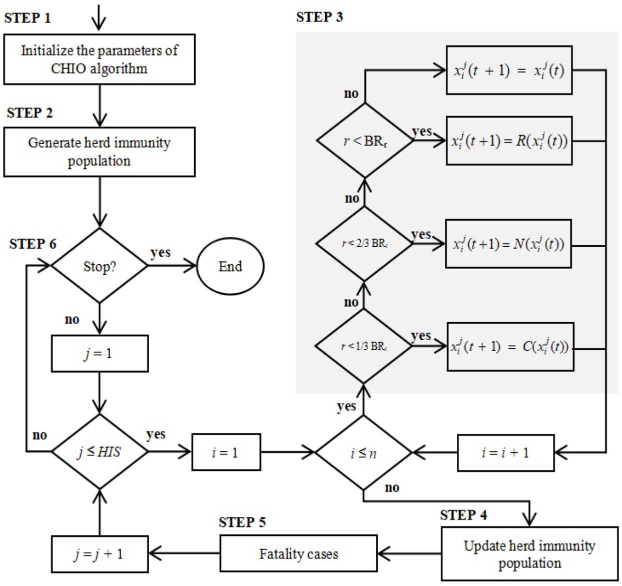


**Figure 1: Flowchart of the Coronavirus Herd Immunity Optimizer (Lifted from Al-Betar et.al, 2020).**

## 3 Chaotic Maps

Chaotic systems are rule-based systems with mathematically accessible input-output relationships. The output behaves randomly yet adhering to specific analytically grounded relationships. Its output is confined to two values making them suited for usage as a random factor to weaken particle movement paths and improve their exploring capabilities [12]. Studies in the past proved to show positive impact of using chaotic maps in optimization algorithms ([13],[14]).
In this study, two types of chaotic maps that have improved previous studies will be presented and used.

### 3.1 Tent Chaotic Map

The Tent Chaotic Map is defined in Equation 11 where $x_k$ is a random value from 0 to 1 and $\mu$ is a value chosen between 0 to 2. The values from this chaotic map ranges from 0 to 1 and tends to generate a tent-like shape of the graph.

$$x_{k+1} = \begin{cases} \mu x_k & x_k < 0.5 \\ \mu(1 - x_k) & x_k \ge 0.5 \end{cases} \qquad (11)$$

A study by Demidova and Gorchakov [14] comparing tent chaotic map to non-chaotic pseudorandom number generators showed that the former can yield more symmetrically and uniformly distributed real numbers. The tent chaotic map was used in place of the traditional pseudorandom number generators in implementing the Fish School Search Algorithm as tested on benchmark test functions. With the incorporation of exponential step into the algorithm, accuracy was improved and was proven to be more effective compared to Particle Swarm Optimization and Genetic Algorithms.

### 3.2 Circle Chaotic Map

The Circle chaotic map or the sine circle map, as seen in Equation 12, is an iterated one-dimensional map [15]. The value $x_k$ is a random number between 0 and 1 and a and b are the external applied frequency and strenghth of nonlinearity, respectively.

$$x_{k+1} = x_k + \mathrm{b} - \frac{a}{2\pi} sin(2\pi x_k) mod(1) \qquad (12)$$

The same study by Demidova and Gorcothakov [14] showed that the circle chaotic map was competitive compared to other chaotic maps. It was, however, less superior than the tent chaotic map.

With the potential of CHIO in solving optimization problems and the proven effectiveness of chaotic maps to improve existing algorithms, this study investigated using the classical CHIO (CPC) and CHIO with Chaotic Maps (CCMPC) to solve the *p-center* problem.

## 4 CPC and CCMPC for the *p-center* problem

This study focuses mainly on, one, the performance of CHIO on *p-center* problems and, two, the effect of replacing the pseudo random number generators for the initial population generation and generation of fatality cases using tent and circle chaotic maps into CHIO. The classic CHIO will be labelled as CPC while any of the two CHIO with chaotic maps with be labelled as CCMPC.

### 4.1 Test Data and Parameter Settings

*4.1.1 Datasets*. The algorithms will be tested using datasets from the cities of Davao region in the Philippines. The Davao City dataset was obtained from the Phil-LIDAR2 project of the Anonymous *(details omitted for double-blind reviewing)* and has 186,244 household data. The Digos City dataset was lifted from the Phil-LIDAR1 project of the same university with a total number of 23,882 household data. Lastly, the geotagged dataset of Tagum City that was obtained from the City Planning Development Office of Tagum City with 60,000 household data.

In the interest of time and limited computing resources during the conduct of the study, the study utilized a sample of 30% of the total household data for each city. Thus, the number of household data used in the study for Davao, Digos, and Tagum datasets were 55876, 7165, and 18074, respectively.

*4.1.2 Parameter Settings*. Presented in Table 1 are the parameter settings including the algorithmic parameters for CHIO, and the parameters used for the chaotic maps. Furthermore, two values of the basic reproduction rate parameter were also experimented on when dealing with slow (BRr = 0.01) and fast-spreading (BRr = 0.90) infections. All in all, there are 18 different parameter configurations considering 3 data sets (Davao, Digos, and Tagum), algorithm versions (Classic Chio, Chio with Circle Chaotic Map and CHIO with Tent Chaotic Map) and basic reproductive rate (0.01 and 0.90). Each parameter configuration is independently run 30 times to allow for statistical analysis.

## 4.2 Experimental Results

The goodness of the performance of CPC and CCMPC is initially gauged by the quality of the fitness values using Equation 1 (descriptive and inferential statistics), and then checks the algorithm performance based on CPU Time (time at which the algorithm first found the best solution) as a secondary criterion.

Table 1. Experimental parameters.

| Parameter | Value |
|---|---|
| Basic reproduction rate (BRr) | 0.01 and 0.90 |
| Maximum age for infected cases (max_age) | 100 |
| Initial infected cases ($C_0$) | 1 |
| Maximum number of iteration (max_itr) | 30000 |
| Total number of solution (HIS) | 30 |
| Number of facilities/centers (n) | 6 |
| μ (Tent map) | 2 |
| *a* (Circle map) | 0.5 |
| *b* (Circle map) | 0.2 |

*4.2.1 Basic reproduction rate 0.01.* Results for the different datasets and algorithms on using BRr = 0.01 are presented in this section. Statistical analyses were done on the fitness values, and results show that no significant difference was observed between the Classic CHIO (CC) and CHIO with Tent Chaotic map (CTC) for the Davao City dataset. Figure 2 (A and B) shows the convergence of the fitness values for these two algorithms as the iteration progresses. There is a drastic decrease/improvement in the fitness values on its first 10000 iterations and has been gradually declining upon further iterations. It is also important to note that improvements are still visible from iteration 25,000 onwards, where a possible decrease in the fitness value may still be achieved if iterations are to be increased.

Table 2 also shows the descriptive statistics associated with this experiment as well as the time at which the best fitness value was first observed (CPU time). Classic CHIO obtained the most desirable solution terms of best, worst, mean fitness values and CPU time. CHIO with Circle Map, however, generated very similar (stable) fitness values as shown by the lowest standard deviation value.

Graphs of the best performing algorithms in each dataset, showing the initial and final position of the centers were overlaid on Google earth. The radius of the circles indicates the fitness value of the solution generated. Final locations tend to be closer to the actual households of the city and are more within the boundary of the region shows a more desirable result. Figure 2 (A and B) shows that the radii of the red circles are smaller than the green circles, indicating improvement on the solutions generated by the algorithm as it progresses.

Statistical analysis for the different algorithms on the Digos City dataset showed that Classic CHIO was significantly better than the two CCMPCs while Classic CHIO and CHIO with Circle chaotic map were not significantly different from each other when tested on the Tagum City dataset.

Table 2. Best, worst, mean, and standard deviation for Davao City dataset using $BR_r$=0.01 in meters.

| | Classic CHIO | CHIO w/ Tent | CHIO w/ Circle |
|---|---|---|---|
| Best Fitness [BF] | **8426.3650** | 8949.5450 | 8790.3260 |
| Worst Fitness [WF] | **12941.3500** | 14457.9340 | 13728.8590 |
| Mean Fitness [MF] | **9429.0290** | 9695.9080 | 9911.0030 |
| Standard Deviation [SD] | 516.7280 | 492.1120 | **424.5360** |
| Mean CPU Time [MCT] (s) | **116406.0644** | 117120.9388 | 124436.3418 |

Value/s in bold character indicates the best value among the setup.

Tables 3 and 4 show the descriptive statistics of the different algorithms for Digos City and Tagum City, respectively. For the Digos City dataset, the Classic CHIO obtained the most desirable solution in terms of best, worst, and mean fitness values. CHIO with Tent Chaotic Map, however, generated very stable fitness values as well as faster CPU time. For the Tagum City dataset, on the other hand, the Classic CHIO was able to find the best fitness value and best mean fitness value. CHIO with Circle Chaotic Map was the most stable as shown by the least standard deviation and the quickest to finish among the three.

Table 3. Best, worst, mean, and standard deviation for Digos City dataset using $BR_r$=0.01 in meters.

| | Classic CHIO | CHIO w/ Tent | CHIO w/ Circle |
|---|---|---|---|
| BF | **2051.7720** | 2299.6306 | 2370.4583 |
| WF | **3393.7780** | 3921.0970 | 5092.3250 |
| MF | **2451.7308** | 2601.8463 | 2660.9530 |
| SD | 151.7184 | **137.1814** | 153.01201 |
| MCT | 15498.9406 | **13063.4300** | 16704.6842 |

Value/s in bold character indicates the best value among the setup.

Table 4. Best, worst, mean, and standard deviation for Tagum City dataset using $BR_r$=0.01 in meters.

| | Classic CHIO | CHIO w/ Tent | CHIO w/ Circle |
|---|---|---|---|
| BF | **5007.1218** | 5166.2807 | 5142.2310 |
| WF | **7586.5770** | 8169.4670 | 9426.3870 |
| MF | **5619.5590** | 5793.7950 | 5757.1058 |
| SD | **220.5652** | 273.1929 | 259.5690 |
| MCT | 36112.5090 | 38435.4468 | **33713.8444** |

Value/s in bold character indicates the best value among the setup.

Shown in Figure 2 (C) and Figure 2 (D and E) are the convergence graph and the initial and final positions for the best performing algorithms for Digos and Tagum datasets. Convergence behavior of the fitness values for the different algorithms as iterations progressed for Digos and Tagum City dataset were similar to that of the Davao City these graphs. It emphasizes that the algorithm can improve solutions even with a different basic reproduction rate, and improvements were evident by a tighter (smaller) radius of *p-center*s at the end of the iteration.

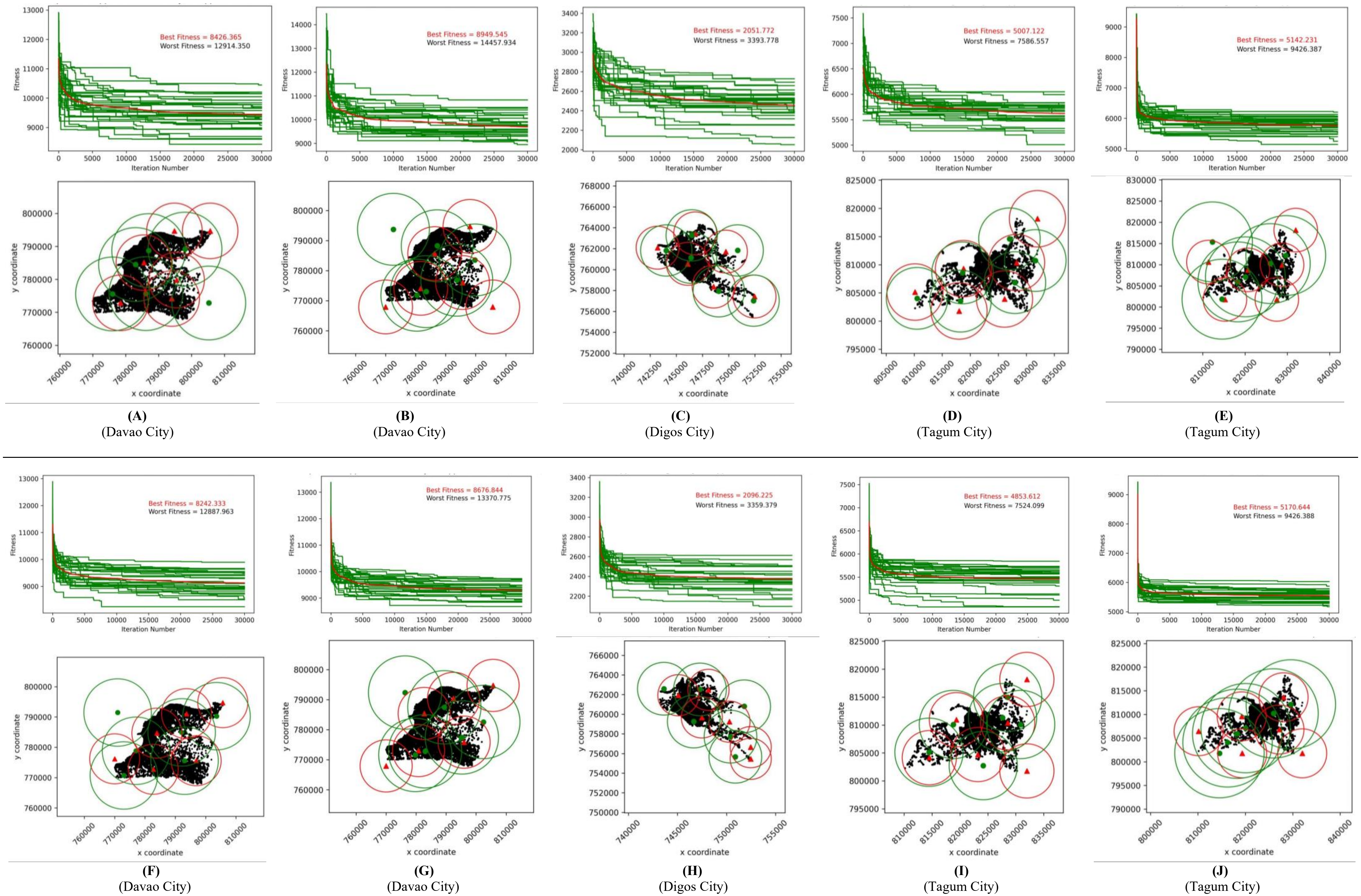


**Figure 2: Convergence Graph(top) and Initial [green] and Final [red] (bottom) locations of the best solution/s using $BR_r$ = 0.01 (A through E) and $BR_r$ = 0.90 (F through J).**

*4.2.2 Basic reproduction rate 0.90.* Results for the different datasets and algorithms on using BRr = 0.90 are presented in this section. It is worth noting that the best algorithms for a particular dataset and behaviour of convergence maps for this BRr had the exact similar results as with BRr = 0.01. The descriptive statistics resulting from BRr = 0.90, however, were different from that of BRr=0.01.

Tables 5, 6 and 7 show the descriptive statistics of the different algorithms for Davao City, Digos City and Tagum City, respectively. For the Davao City dataset, Classic CHIO acquired the most desirable value for the best and mean fitness, while CHIO with Tent Chaotic Map has the lowest worst fitness value. CHIO with Circle Chaotic Map, however, has least deviated fitness and the fastest to converge to the best solution. For the Digos City dataset, Classic CHIO obtained the most appealing results for the best, worst, and mean fitness values, as well as the most stable fitness as depicted from it having the lowest standard deviation among the three. All while, CHIO with Circle Chaotic Map is the quickest to arrive with the best solution. Lastly, for the Tagum City dataset, the best fitness value among the algorithms was acquired by CHIO with Tent Chaotic Map as well as the quickest to arrive at the best solution. While Classic CHIO has the best result for the values of the worst and mean fitness. CHIO with Circle Chaotic Map, however, is the most stable having the least deviated fitness values among the algorithms.

Statistical analysis for the different algorithms using BRr=0.90 for the different datasets produced similar results as that of the BRr=0.01. That is, Classic CHIO and CHIO with Tent Chaotic Map for the Davao City dataset; Classic CHIO for the Digos City dataset and Classic CHIO and CHIO with Circle Chaotic Map for the Tagum City dataset.

Plots of the solutions of the best performing algorithms for each dataset and its corresponding graphs are presented in Figure 2 (F through J). Convergence graphs for all these figures showed similar in behaviour as observed with BRr=0.01 where clear decrease in convergence trend can be observed. Plots of the *p-center*s also show improvement with the final (red) locations having tighter radii than the initial (green) locations. Again, it is to note that some final locations were in a body of water as seen in Figure 2 (F through J).

Table 5. Best, worst, mean, and standard deviation for Davao City dataset using $BR_r$=0.90.

| | Classic CHIO | CHIO w/ Tent | CHIO w/ Circle |
|---|---|---|---|
| BF | **8242.3331** | 8676.8444 | 8690.9238 |
| WF | 9893.9472 | **9734.0202** | 9893.1809 |
| MF | **9125.2997** | 9291.7594 | 9406.9496 |
| SD | 412.7521 | 292.3905 | **291.1344** |
| MCT | 129340.2153 | 117904.6340 | **93334.1110** |

Value/s in bold character indicates the best value among the setup.

Table 6. Best, worst, mean, and standard deviation for Digos City dataset using $BR_r$=0.90.

| | Classic CHIO | CHIO w/ Tent | CHIO w/ Circle |
|---|---|---|---|
| BF | **2096.2230** | 2153.8885 | 2252.4427 |
| WF | **3359.3790** | 3925.4880 | 5092.3180 |
| MF | **2373.0290** | 2541.3992 | 2586.2832 |
| SD | **123.0010** | 178.1307 | 152.7004 |
| MCT | 15309.2362 | 12273.7748 | **11485.1806** |

Value/s in bold character indicates the best value among the setup.

Table 7. Best, worst, mean, and standard deviation for Tagum City dataset using $BR_r$=0.01.

| | Classic CHIO | CHIO w/ Tent | CHIO w/ Circle |
|---|---|---|---|
| BF | 4853.6119 | **4811.3694** | 5170.6444 |
| WF | **7524.0990** | 8061.8010 | 9426.3880 |
| MF | **5460.7795** | 5645.6838 | 5540.4572 |
| SD | 247.0989 | 264.1345 | **199.0190** |
| MCT | 37433.6903 | **30490.1096** | 32811.7169 |

Value/s in bold character indicates the best value among the setup.

*4.2.3 BRr 0.01 vs BRr 0.90.* For better understanding of the comparisons on this section of the results, nine experimental setups were defined. Classic CHIO algorithm was used for the setups S1, S2, and S3, using Davao, and Tagum City datasets, respectively. Additionally, CHIO with Tent Chaotic Map, and with Circle Chaotic Map, were used for S4 through S6, and S7 through S9, respectively using the same dataset order as classic CHIO.

Corresponding best fitness values were recorded for each of the experimental setup and is presented using Table 9 for the different basic reproduction rates. It can be observed that 7 of the 9 setups performed better when the basic reproduction rate is 0.90.

Table 10 shows the summary of the statistical result using the best fitness values for all the algorithms. At a significance level of 0.05, 8 of the 9 setups have enough reason to believe that there is a significant difference between the two basic reproduction rates in terms of fitness value.

Table 9. Best fitness values and Mean Ranks using basic reproduction rates 0.01 and 0.90.

| Setup | Fitness Value | | Mean Rank | |
|---|---|---|---|---|
| | BRr = 0.01 | BRr = 0.90 | BRr = 0.01 | BRr = 0.90 |
| S1 | 8426.3654 | **8242.3331** | 28.5000 | **32.5000** |
| S2 | **2051.7720** | 2096.2250 | 31.1000 | **29.9000** |
| S3 | 5007.0532 | **4853.6119** | 29.7300 | **31.2700** |
| S4 | 8949.5453 | **8676.8444** | 31.8300 | **29.1700** |
| S5 | 2299.6306 | **2153.8885** | 30.8300 | **29.1400** |
| S6 | 5166.2807 | **4811.3694** | 35.7700 | **25.2300** |
| S7 | 8790.3264 | **8690.9238** | 36.7000 | **24.3000** |
| S8 | 2370.4583 | **2252.4427** | 37.9300 | **23.0700** |
| S9 | **5142.2311** | 5170.6444 | 30.8000 | **30.2000** |

Value/s in bold character indicates better value for fitness and mean rank comparing two basic reproduction rates.

Table 10. Mann-Whitney U-test result using at alpha = 0.05.

| Setup (0.01 vs 0.90) | *p*-value |
|---|---|
| S1 | **0.018737** |
| S2 | **0.011710** |
| S3 | **0.018005** |
| S4 | **0.001030** |
| S5 | 0.074779 |
| S6 | **0.000604** |
| S7 | **0.000604** |
| S8 | **0.000604** |
| S9 | **0.000604** |

Value/s in bold character indicates significant difference.

Furthermore, the mean ranks of each comparison are also recorded and are also presented in Table 9. It can be observed that best fitness values using the basic reproduction rate value 0.90 performs better compared to basic reproduction rate value 0.01.

Based on the descriptive and inferential statistical evidence, using basic reproduction rate value of 0.90 provides better results. A lager basic reproduction rate (BRr) means that the generated solutions underwent frequent changes (refer to Equation 3), providing a larger solution space to explore.

## 5 Conclusion

CHIO showed a good balance between exploration and exploitation as concluded by Al-Betar et.al. With this and the algorithm being new, it presented itself as a good area for research to further explore its capabilities, hence, being used in this study to solve the *p-center* problem.

When tested, Classic CHIO resulted in improvement on the fitness values of the solution generated, indicating that the population has been acquiring herd immunity. This means the effectiveness of CHIO in solving the *p-center* problem.

Overall, any of the algorithms (CPC and CCMPC) are effective for all tested datasets. From which can be concluded from the statistical results on Davao and Tagum City showing no significant difference between algorithms. However, CPC emerged to be one if not the lone best performing algorithm for all three datasets on all BRr experimental values.

The value of the BRr also affects how the algorithms search the solution space. The larger BRr, in this case 0.90, the better algorithm performs compared to 0.01 with respect to the best fitness. The increased rate of exploring the solution space could have affected the results.

Further investigation on parameters used, and feasibility of the location of the centers still calls for improvement. The resulting location of the centers can also be subjected to land use land cover of each city to determine viability of putting up centers in those areas. This can also include elevation, nearness to highway, among others. The authors are also currently investigating the algorithm using full household data set used in this study.

The extent of the capabilities of this algorithm are still to be tested and can also be used to solve other optimization problems. However, the methodology used in this study can already aid local government units in finding best initial locations for services such as emergency centers, early warning devices, fire stations, bicycle stations, and others.

Nevertheless, CHIO was found to be very capable and can be considered as an alternative to state-of-the-art methods in solving the *p-center* problem.

## ACKNOWLEDGMENTS

The authors wish to acknowledge Anonymous for providing invaluable computational resources. The authors also wish to extend their sincerest gratitude to the ff: Anonymous for allowing the use of the real household datasets. *Details omitted for double-blind reviewing.*

## REFERENCES

[1] M Taghavi and H Shavandi (2012). The P-center Problem under Uncertainty. Journal of Industrial and Systems Engineering. 6 (1), 48-57.
[2] A Kaveh and H Nasr (2011). Solving the conditional and unconditional p-center problem with modified harmony search: A real case study. Scientia Iranica. 18 (4), 867-877.
[3] H Fayed and A Atiya (2012). A mixed breadth-depth first strategy for the branch and bound tree of Euclidean k-center problems. Computational Optimization and Applications. 54, 675-703.
[4] E Gaar and M Sinnl (2022). A scaleable projection-based branch-and-cut algorithm for the p-center problem. European Journal of Operations Research. 303, 78-98.
[5] C Contardo, M Iori and R Kramer (2019). A scaleable exact algorithm for the vertex p-center problem. Computers and Operations Research. 103, 211-220.
[6] J Nematian and M Sadati (2015). New methods for solving a vertex p-center problem with uncertain demand-weighted distance: A real case study. International Journal of Industrial Engineering Computations. 6 (2), 253-266.
[7] Anonymous (2019). Details omitted for double-blind reviewing.
[8]. AH Yin, TQ Zhou, JW Ding, QJ Zhao and ZP Lv (2017). Greedy Randomized Adaptive Search Procedure with Path-Relinking for the Vertex p-center problem. Journal of Computer Science and Technology. 32, 1319-1334.
[9] Anonymous (2022). Details omitted for double-blind reviewing.
[10] MA Al-Betar, ZAA Alyasseri and MA Awadallah (2020). Coronavirus Herd Immunity Optimizer. Neural Computing and Application. 33, 5011-5042.
[]11] C Kumar and T Gunasekar (2021). MOCHIO: A Novel Multi-Objective Coronavirus Herd Immunity Optimization Algorithm for Solving Brushless Direct Current Wheel Motor Design Optimization Problem. Journal for Control, Measurement, Electronics, Computing and Communications. 63 (1), 149-170.
[12] M Mobaraki, R Boostani and M Sabeti (2020). Chaotic-based Particle Swarm Optimization with inertia weight for optimization tasks. Journal of AI and Data Mining. 8 (3), 303-312.
[13] H Lu, X Wang, Z Fei, and M Qiu (2014). The Effects of Using Chaotic Map on Improving the Performance of Multi Objective Evolutionary Algorithms. Mathematical Problems in Engineering. 2014, 1-16.
[14] LA Demidova and AV Gorchakov (2020). A study of Chaotic Maps Producing Symmetric Distributions in Fish School Search Optimization Algorithm with Exponential Step Decay. Symmetry. 12 (5), 784- 802.
[15] GC Layek (2015). An introduction to dynamical systems and chaos. Springer, New Delhi, India.