American Journal of Artificial Intelligence
2022; 6(1): 10-19
http://www.sciencepublishinggroup.com/j/ajai
doi: 10.11648/j.ajai.20220601.12
ISSN: 2639-9717 (Print); ISSN: 2639-9733 (Online)

Artificial Corona Algorithm to Solve Multi-objective Programming Problems

Alia Youssef Gebreel

Operations Research, Cairo University, Cairo, Egypt

Email address:
y_alia400@yahoo.com

To cite this article:
Alia Youssef Gebreel. Artificial Corona Algorithm to Solve Multi-objective Programming Problems. *American Journal of Artificial Intelligence.* Vol. 6, No. 1, 2022, pp. 10-19. doi: 10.11648/j.ajai.20220601.12

Received: February 12, 2022; **Accepted**: March 18, 2022; **Published**: March 31, 2022

Abstract: Multi-objective optimization is a branch of mathematics used in a large range of applications. It deals with optimization problems involving two or more conflicting objective functions to be optimized. Consequently, there is not a single solution that simultaneously optimizes these objectives, but a set of compromise solutions. These compromise solutions are also called non-dominated, Pareto-optimal, efficient, or non-inferior solutions. The best solution of this set is the one closest point to the utopia point. There are several approaches to perform multi-objective optimization. Undoubtedly the future of multi-objective optimization programming is in artificial intelligence applications. One of the artificial intelligence models is the Corona algorithm. It aims to simulate the epidemic behavior of the Corona virus that affects people's health and its treatment. In this paper, the artificial Corona algorithm is introduced and expanded for solving multi-objective programming problems, in which other models are not effective. The algorithm operates by iteratively selecting the initial values for decision variables of a multi-objective programming problem. The values of objective functions and constraint(s) are calculated. This proposed approach depends on a linear formula to update the solution. An acceptable efficient solution that has a minimum distance value from the utopia point is selected as the best point. To demonstrate the effectiveness of the proposed approach, some illustrative examples are given. These examples include both linear and nonlinear problems. The results indicate that the proposed approach has a high speed and capability to obtain the best solution when compared with other similar works of literature.

Keywords: Artificial Corona Algorithm, Multi-objective Problems, Best Solution

1. Introduction

Over 25 years ago, there was an increasing amount of literature on multi-objective optimization problems using artificial intelligent approaches. Nowadays, since the spread of the Corona virus (COVID-19) in the world, many researchers have been turned to their research for studying this virus and simulating its behavior in various scientific aspects.

One example of them is the artificial Corona optimization algorithm, which was put forward by Gebreel [1] uses a linear formula to get the optimal solution based on three real numbers $(0.5, \sqrt{0.5}, 1.0)$. It uses deterministic operators such as the most classical optimization methods, unlike stochastic operators used in evolutionary optimization algorithms. Moreover, this procedure doesn't require any derivative information. Therefore, it is flexible and simple to implement with high accuracy.

Figure 1. *Corona virus (COVID-19) around the world.*

A multi-objective problem has more than one objective function that conflict with each other to be minimized or

maximized simultaneously. So, there doesn't typically exist an optimal solution that minimizes or maximizes all objective functions simultaneously, but instead, it is represented by the best tradeoff between competing objectives in the objective space [2, 3]. It's known that multi-objective optimization involves two search spaces: The decision variable space and the objective or criterion space. In any optimization approach, the search is performed in the decision variable space. Then, the proceedings of an approach in the decision variable space can be traced in the objective space. In constrained problems, the objective space and decision space present feasible and infeasible regions [4-9].

There are large numbers of researchers dealing with multi-objective optimization programming problems in both optimization ways. One of the most widely used methods is called the traditional methods. But, the other researches are called the heuristic approaches. Most of these researches presented a set of efficient solutions in solving such optimization problems. Some of them are concerned with providing the preferred or best efficient solution in the subset of applications, such as De & Yadav [10], Gupta *et al.* [11], Upmanyu & Saxena [12], Wang & Rangaiah [13], Gebreel [3, 14-16], and Kamal *et al.* [17]. It is necessary to find a new way to produce an accurate best point for both convex and non-convex or linear and nonlinear multi-objective problems within reasonable time.

Corona approach (CA) will become the major alternative for the other heuristic models in the area of multi-objective optimization problems because of its faster convergence rate and higher accuracy.

This paper attempts to obtain the best efficient solution of multi-objective programming problems using the artificial Corona algorithm. The following section begins work with preliminaries about this study. After that, the proposed algorithm is introduced to solve multi-objective problems in section 3. Then, section 4 is devoted to illustrative some examples to demonstrate the solution procedures. Finally, section 5 gives conclusions and future work.

2. Preliminaries

A multi-objective programming (MOP) problem can be stated as follows:

(MOP):

Minimize/ Maximize:

$$F(x) = (f_1(x), f_2(x),..., f_k(x)), k \geq 2,$$

Subject to:

$$S = \{x \in R^n / g_m(x) \leq 0, r=1, 2, \ldots, M,$$

$$h_l(x) = 0, l=1, 2,..., L\}. \quad (1)$$

Where:

$F(x) = (f_1(x), f_2(x),..., f_k(x))$ is a vector of k objective functions, and k is used to identify the number of objective functions,

$x= (x_1, x_2, \ldots, x_n)^T$ is a set of decision variables,

n is a number of decision variables,

T is the transpose operator,

The set S is a non-empty and feasible region included in R^n that is determined by the constraints on the multi-objective problem. It is defined as $S=\{x \mid x \in R^n, g_m(x) \leq 0, m= 1, 2, \ldots, M, h_l(x)= 0, l=1, 2, \ldots, L\}$,

The inequality $g_m(x)$ and equality $h_l(x)$ are real valued functions defined on S,

M and L are the numbers of inequality and equality constraints, respectively.

In the following sections, some terminologies are used in this paper:

2.1. Efficient Solution

A decision vector $x^* \in S$ is said to be an efficient solution if there does not exist another decision vector $x \in S$ such that $f_i(x) \leq f_i(x^*)$ for i = 1, 2, …, k and $f_j(x) < f_j(x^*)$ for at least one index j [2].

2.2. Utopia (Ideal) Point

The point $(f_1(x_1^*), f_2(x_2^*),..., f_k(x_k^*))$ in the objective space is called *utopia (ideal) point* [14].

2.3. The Best Efficient Point

The best efficient solution on the efficient set is a feasible solution that has the shortest distance to the utopia solution.

2.4. Euclidean Distance Formula

The Euclidean distance formula is used to find the distance between two points on a plane [17, 18]. This formula calculates the distance between two points (x_1, y_1) and (x_2, y_2) as follows:

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}. \quad (2)$$

The distance between two points is the length of the path connecting them. The shortest path distance is a straight line. In *k* dimensional space, the Euclidean distance formula is:

$$D = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}. \quad (3)$$

3. The Proposed Algorithm

Corona virus (CV) was explored, and it is an infectious disease that spreads throughout the world. This virus can infect people, and these people may die, infect other people, or recover after the infection. Therefore, the main goal is to get rid of this virus and improve people's health, which corresponds to the best solution for a given multi-objective optimization problem. The key idea of the proposed algorithm is to solve such problem iteratively for providing the best efficient solution. The used measure is the minimum Euclidean distance from the ideal point to find an efficient solution at each iteration. The procedure continues until the best efficient solution is satisfied with the current feasible solution and does not want to change the multi-objective values anymore. A multi-objective optimization problem may contain some of constraints, which any feasible solution

(including all optimal solutions) must be satisfied.

In the following sections, a brief overview of the Corona search algorithm and its modification procedures to solve multi-objective programming problems are introduced.

3.1. Overview of the Corona Search Algorithm

The Corona search algorithm is an artificially intelligent algorithm that is inspired by the Corona virus to solve optimization problems efficiently. First, an initial value for each decision variable is selected based on any one of the three values $(0.5, \sqrt{0.5}, 1)$. Second, the new Corona vector $(x= x_1, x_2, \ldots, x_n)^T$ is generated based on the linear formula:

$$x_{i+1}= a_1 x_i + a_2, \quad (4)$$

Where: a_1 and a_2 are real numbers and they consider as spreading rates of Corona virus (controlled parameters).

The setting values of decision variables and parameters are inspired by three cases of the Corona virus: Infection of three parts of the human body (nose, throat, and respiratory), treatment, or/and death.

If the new solution is better than the worst one in the Corona memory (CM), the new solution is included in the memory instead of the worst one. The algorithm continues to run until convergence to the required solution, or the maximum number of iterations is reached [1].

In multi-objective optimization problems, the aim is to find a set of non-dominated solutions and after that, the decision-makers select one preferred solution for them based on their qualitative experiences. But, CA deals with such problems to find the best non-dominated solution step by step, as shown in the following sections. There is no doubt that it is designed to be simple and fast computing. Through this process, time is saved for achieving the best solution.

3.2. The procedure of Corona Approach to Solve Multi-objective Programming Problems

The main steps to implement the proposed approach can be stated as follows:

Step 1: Initialize the problem by Corona approach.

Step 2: Prepare a memory vector.

Step 3: Update the Corona memory.

Step 4: Check the stopping criterion.

These steps are described in the next four subsections.

3.2.1. Initialize the Problem by Corona Approach

The Corona approach begins its search with an initial solution within a specified lower and upper bound for each variable, as follows: $x_i^L \le x_i \le x_i^U$, i = 1, 2, …, n. The lower bound usually depends on one of the three real numbers $(0.5, \sqrt{0.5}, 1.0)$ for each decision variable of a multi-objective problem. But the upper bound for each variable is calculated by putting all variables at zero values in the set of constraints except one variable "x_i". Then the upper bound has a maximum value of x_i.

3.2.2. Prepare a Memory Vector

A group of decision variables for a multi-objective problem is stored at first in a vector called the Corona memory. CM is the core part of the Corona search approach. After that, the problem constraints ($g_1(x)$, $g_2(x)$, …, $g_M(x)$, $h_1(x)$, $h_2(x)$, …, $h_L(x)$) and the objective (fitness) functions are stored next to these decision variables. In this step, the Corona vector is initially filled with any one of the three values $(0.5, \sqrt{0.5}, 1.0)$. The corresponding values of objective functions and constraints are calculated and stored in CM as follows:

$$CM=[x = x_1, x_2, \ldots, x_n, g_1(x), g_2(x), \ldots, g_M(x), h_1(x), h_2(x), \ldots, h_L(x), f_1(x), f_2(x),..., f_k(x)]^T. \quad (5)$$

3.2.3. Update the Corona Memory (CM)

The new Corona vector $(x= x_1, x_2, \ldots, x_n)^T$ is generated based on the Corona formula $(x_{i+1}= a_1 x_i + a_2)$. If the new solution is better than the previous one in the CM; judged in terms of the distance of objective functions from the utopia point and feasibility of constraints (if any); it is included in the CM, and the existing worst solution is excluded from the CM.

3.2.4. Check the Stopping Criterion

The calculating Euclidean distance of an obtained solution depends on determining the utopia point, which optimizes individually each of the objective functions.

The process of creating a new solution from the CM and updating memory may be called an iteration. After each iteration of the proposed approach:

1. If the best efficient solution is met, stop.
2. Otherwise, repeat Steps 2 and Steps 3 until the stopping criterion (or the maximum number of iterations that is determined arbitrarily) is satisfied, and computation is terminated.

All of these steps are illustrated using pseudo-code as in the following *Figure 2*:

CV Approach

Begin

1- Initialize the CVA:

Define fitness objectives **(**F(x) = ($f_1(x)$, $f_2(x)$, ... , $f_k(x)$), $k \ge 2$**)**,
x= **(**$x_1, x_2, \ldots, x_n$**)^T**, and the constraints (if any) = $g_m(x)$ and $h_l(x)$,
Define "a_1", and "a_2",
Define the maximum number of iterations (NI).

2- Generate a new solution:

Selecting an initial point from these three values: $(0.5, \sqrt{0.5}, 1)$,
with taking into consideration that:
Max values of x_i= the upper bound (x_i^U) of x_i, i= 1, …, n.

3- Calculate the fitness functions:

while (*Max number of iterations* < NI) do
  while (V_i < *number of variables*) do
    Use the selected point to calculate Euclidean distance
    of fitness functions from the utopia point
      if the new solution is better than the worst solution
      then replace the worst solution by the new one with
        update the Corona memory based on:
        $x_{i+1}= a_1 x_i + a_2$
      end if
  end while
end while
Best efficient solution = find the current efficient solution that has the minimum distance to the utopia point

End

Figure 2. *Pseudo-code of CV approach.*

The flowchart of the algorithm execution in a multi-objective optimization problem is given in *Figure 3*:

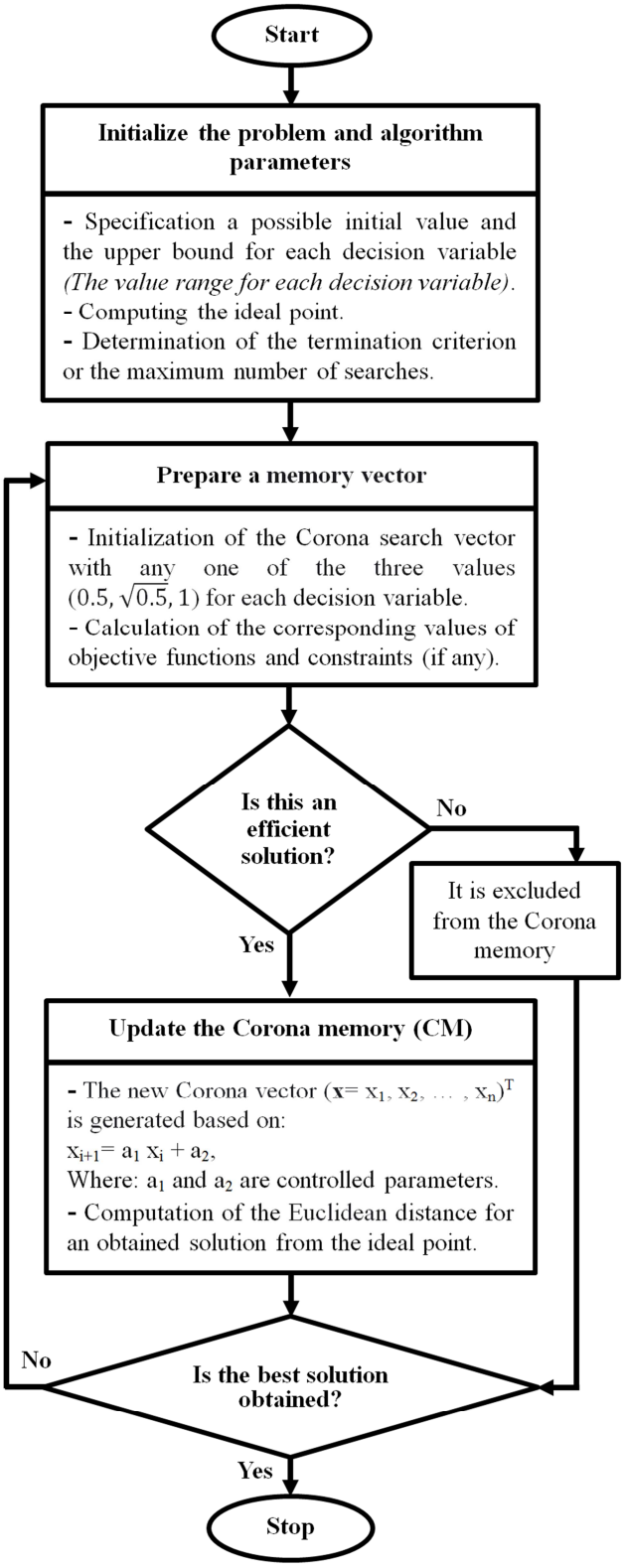


Figure 3. *A flowchart of the procedure of Corona approach to solve multi-objective programming problems.*

The following statement can be proved:

In the objective space of a multi-objective programming problem, a solution x^{**} for the proposed algorithm is said to be the best efficient solution if and only if there doesn't exist another efficient solution x such that:

$$\text{Distance } (x) > \text{distance } (x^{**}). \tag{6}$$

Proof:

The proof comes from the assumption that the solution x^{**} to the Corona approach is not the best efficient and then shows that this assumption violates the definition of the best efficient solution. This means that x^{**} has a minimum distance from the utopia point.

It is an important relationship between the optimum value to the original MOP problem and its optimum value that is achieved by the best point for this problem.

Corollary:

In the minimum (maximum) case, the optimum value of any multi-objective programming problem (without weights) is always less (more) than or equal to the corresponding optimum value of the solved problem to get the best efficient solution.

Proof:

In the minimum case, consider x^* is the optimal solution to the original MOP problem, and x^{**} is the best efficient solution for this problem. Let $\bar{x}$ be any other efficient solution to the MOP problem, and it lies between two solutions: x^* and x^{**} such as:

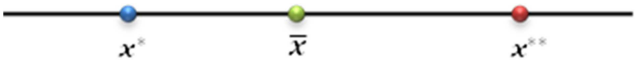


Figure 4. *Three efficient solutions for a multi-objective problem.*

Based on lines and line segments definitions [19], it leads to:

$$F(x^*) \leq F(\bar{x}). \tag{7}$$

Since the three solutions are located at the same line segment, then

$$\bar{x} = (1- W)\, x^* + W\, x^{**}, \tag{8}$$

Where W is a real number with $0 \leq W \leq 1$, and

$$F(x^*) \leq F(\bar{x}) \leq (1- W)\, F(x^*) + W\, F(x^{**}), \tag{9}$$

$$F(x^*) \leq F(x^*) - W\, F(x^*) + W\, F(x^{**}), \tag{10}$$

$$F(x^*) \leq F(x^{**}). \tag{11}$$

Thus the result holds.

In the same way, it can prove in the maximum case that:

$$F(x^*) \geq F(x^{**}). \tag{12}$$

3.3. The Main Advantages of CV Approach

CV approach attempts to find the best efficient solution for multi-objective programming problems and has the following advantages:

1. CV approach searches the space of a problem efficiently.

It can converge to the final best efficient solution in multi-objective linear and nonlinear programming problems.
2. This algorithm obtains the best point for both conflicting and non-conflicting multi-objective problems.
3. The effective parameters (a_1 and a_2) in its formula directly affect the speed of Corona's search. Besides, it doesn't have many mathematical requirements related to optimization problems. Accordingly, there is a guarantee that the CV approach converges to the best point due to the nature of its linear formula.
4. The transformation of multi-objective to single-objective optimization doesn't need in this approach for handling multiple objectives (i.e., the optimization of a multi-objective programming problem cannot use the scalarization of objectives). All objectives are treated separately. Therefore, it saves some computational time.
5. The Corona approach is an optimization tool to guide the search towards a set of efficient solutions through searching for the best efficient solution. It can execute by Excel, MATLAB programming, or any other computational software. It can end after a few or several iterations with an accurate solution.
6. It uses deterministic transition rules as most of the conventional methods rather than derivatives or probabilistic transition rules.

Remarks:

1. The starting step of the Corona approach is to minimize (maximize) the objectives by an initial solution, and try to achieve decreasing in the distance from the utopia point by the values of a_1, and a_2, with satisfying the constraints (if any).
2. The Corona approach's performance depends on the appropriate selection of values for the decision variables and parameters (a_1 and a_2) of its linear formula.
3. It is worth mentioning that the best efficient solution is obtained without using the weighting method.
4. When a multi-objective problem is minimized (maximized) without weights, its value is lower (greater) than or equal to the value of such objectives with the best point.
5. The key difference between the CV optimization algorithm and the CV approach to get the best efficient solution is the way of evaluating a new solution. In the optimization process, the optimal solution is selected based on the value of an objective function. But on solving a multi-objective problem, the resulted efficient solution is evaluated based on its distance from the ideal point. Of course, the condition of feasibility must be available in both cases.
6. Corona approach generates a new solution depending on CM and Corona formula. If the new solution strictly dominates the worst solution, the new solution is replaced by the worst solution in the CM.
7. Throughout the search process, the Corona algorithm provides several efficient solutions to a given multi-objective problem, which are very useful to decision-makers in selecting the preferred solution. Furthermore, in the presence of constraint(s), it may achieve an infeasible solution in such a search of the algorithm. Thus after one run, there are three cases: (a) Infeasible solution, (b) Feasible solution but not the best solution, (C) Feasible and the best solution (The best efficient solution). *Figure 5* shows schematically the principles of solving multi-objective optimization based on the Corona approach.

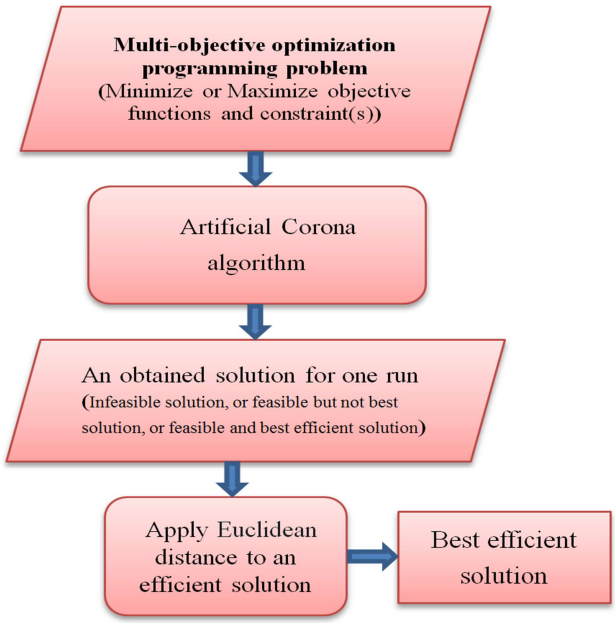


Figure 5. *A schematic of a multi-objective optimization procedure based on the artificial Corona algorithm.*

4. Illustrative Examples

In this section, some examples of several multi-objective optimization problems are presented to show the validity and effectiveness of the artificial Corona approach. The given examples are previously investigated in other works of literature. These examples will be solved by Excel software.

Example 1:

Consider the multiple-objective linear programming problem. It consists of minimizing two linear objectives with two decision variables and two bounded constraints as follows:

$$\text{Min: } (f_1(x) = x_1,\ f_2(x) = (1+x_2) / x_1),$$

$$\text{Subject to: } 0.1 \le x_1 \le 1,$$

$$0 \le x_2 \le 5.$$

This problem has been solved by Gupta & Kumar [20] and Gebreel [16] using three different artificial intelligent algorithms: Genetic algorithm (GA), bacterial foraging optimization (BFO), and harmony search algorithm (HSA).

The steps to solve this example by applying the proposed approach are:

1. Initialize the problem by Corona approach:

It can be starting with an initial solution as (x_1= $\sqrt{0.5}$, x_2= 0.5)^T, and substituting into the two objectives with satisfying the inequalities of the problem. Thus, the corresponding values of the objective functions are 0.7071 and 2.12134, respectively. That leads to the Euclidean distance (D) =1.275.

2. Prepare a memory vector:

Firstly, x_1 and x_2 are set in the Corona memory. Secondly, the corresponding objective functions and the Euclidean distance are stored in CM, as shown in *Table 1*.

3. Update the Corona memory (CM):

The new Corona vector (x_1, x_2)^T is generated based on the Corona formula: x_1= (1.0 × 0.7071 + 0.0331672)= 0.7402672 and x_2= (0 × 0.5 + 0.0)= 0.0, which satisfy the two bounded constraints. The corresponding objective functions are f_1^*= $f_1^*(x)$= 0.7402672, and f_2^*= $f_2^*(x)$= 1.35086358 with the distance of objective functions from the utopia point is 0.73.

4. Check the Stopping Criterion:

As mentioned in the third step, the solution doesn't need any change. Therefore, it is considered to be the best solution to this problem.

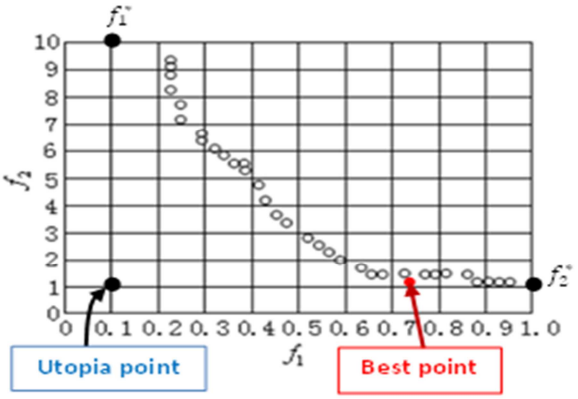


Figure 6. *Experimental results for example (1).*

Optimization effects depend on the initialization of the Corona vector, and two parameters (a_1 and a_2), which control the search process. The process is continued until the obtained solution is feasible and has the minimum distance from the utopia point.

The calculations are shown in *Table 1*. They give the exact minimum distance of objectives from the utopia point of this problem. In addition, it is obvious from *Figure 6* that the result obtained using the Corona search approach is better than those reported previously in the paper by Gupta & Kumar [20]. But, it gives the same results of GA, BFO, and HSA by Gebreel [16] with convergence much faster than them through the optimization process.

Table 1. *The best point of example (1) by Corona approach.*

Data	Initial solution	Update solution based on: $x_{i+1}= a_1 x_i + a_2$
x_1	$\sqrt{0.5}$	1.0 × 0.7071 + 0.0331672 = 0.7402672
x_2	0.5	0.0 × 0.5 + 0.0 = 0.0
f_1	0.7071	0.7402672
f_2	2.12134	1.35086358
f_1+ f_2	2.82844	2.09113078
D	1.275137	0.7301009

It is worth noting that the optimum value of the original problem= 2.0 is lower than the optimum value of the solved problem to get the best point (2.09113078).

Example 2:

The considered multi-objective linear programming problem presented by Sitarz [21] has the following form:

$$\text{Min: } (f_1(x) = -x_1,\ f_2(x) = -x_2),$$

Subject to:

$$x_1 + x_2 \le 8,$$

$$2x_1 + x_2 \le 12,$$

$$x_1 + 2x_2 \le 14,$$

$$9x_1 + 7x_2 \le 63,$$

$$-4x_1 + 10x_2 \le 61,$$

$$2x_1 - x_2 \le 8,$$

$$14x_1 + 3x_2 \le 72,$$

$$x_1, x_2 \ge 0.$$

Figure 7 shaded facets constitute the feasible region. The obtained non-dominated solutions and the best-selected solution are shown in *Figure 8* for two objectives. The utopia point is calculated as: (f_1^*= -4.8, f_2^*= -6.5).

The optimization problem stated before can be solved rapidly in just three iterations after the initial step with the Corona approach. Where the initial value of decision variables is: x_1= x_2= 1, and the upper bounds are: x_1 =14, and x_2= 24, respectively.

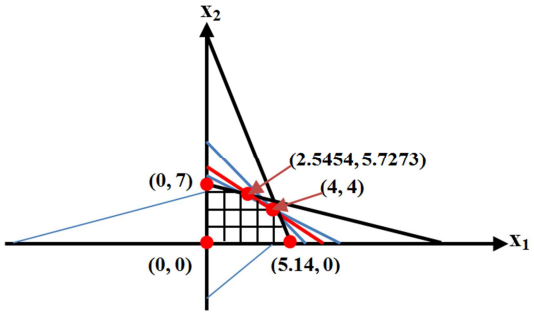


Figure 7. *The feasible region in the decision space of example (2).*

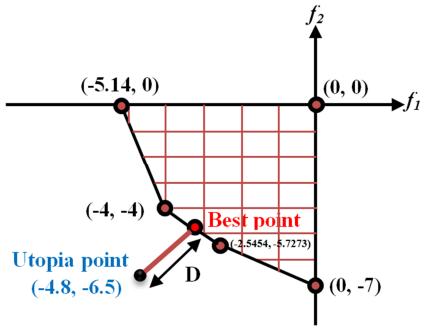


Figure 8. *The best efficient solution in the objective space of example (2).*

As shown in *Table 2* and *Figure 8*, the values of objective functions are equal to the negative values of the decision

variables. Due to the deterministic nature of the Corona approach, the best compromise solution is obtained as f_1^*= -3.15, f_2^*= -4.85. The first, second, third, fourth, fifth, sixth, and seventh constraints are C_1= 8.0, C_2= 11.15, C_3=12.85, C_4= 62.3, C_5= 35.9, C_6= 1.45 and C_7= 58.65, respectively. The Euclidean distance of objectives from the utopia point is D = 2.33 within reasonable execution time. Its value of objectives = the optimum value of the original problem = -8.0. The optimal solution is x_1^*=3.5 and x_2^*= 4.5. The corresponding values of objective functions are -3.5 and -4.5, respectively.

Table 2. *The best point of example (2) by Corona approach.*

Data	Initial solution	Update solution		
		The first iteration	The second iteration	The third iteration
x_1	1.0	3 × 1.0 + 0.0 = 3.0	1.0 × 3.0 + 0.2 = 3.2	1.0 × 3.2 - 0.05= 3.15
x_2	1.0	5 × 1.0 + 0.0 = 5.0	1.0 × 5.0 - 0.2 = 4.8	1.0 × 4.8 + 0.05= 4.85
C_1	2.0	8.0	8.0	8.0
C_2	3.0	11.0	11.2	11.15
C_3	3.0	13.0	12.8	12.85
C_4	16.0	62	62.4	62.3
C_5	6.0	38.0	35.2	35.9
C_6	1.0	1.0	1.6	1.45
C_7	17	57.0	59.2	58.65
f_1	-1.0	-3.0	-3.2	-3.15
f_2	-1.0	-5.0	-4.8	-4.85
f_1+ f_2	-2.0	-8.0	-8.0	-8.0
D	6.685057965	2.343074903	2.33452350599	2.33345237792

Example 3:

This problem was presented by Rafei *et al.* [22] and Gebreel [16]. It was solved by TOPSIS for the convex non-linear multi-objective problems, genetic algorithm, bacterial foraging algorithm, and harmony search algorithm. The problem's formulation is as follows:

$$\text{Max } f_1(x) = x_1^2 + x_2^2 + x_3^2,$$

$$\text{Max } f_2(x) = (x_1 - 1)^2 + x_2^2 + (x_3 - 2)^2,$$

$$\text{Min } f_3(x) = 2x_1 + x_2^2 + x_3,$$

Subject to:

$$g_1(x) = -x_1 + 3x_2 - 4x_3 + 6 \geq 0,$$

$$g_2(x) = -2x_1^2 - 3x_2 - x_3 + 10 \geq 0,$$

$$x \in R^3,\ 0 \leq x_1 \leq 3,\ 0 \leq x_2 \leq 4,\ 0 \leq x_3 \leq 2.$$

The first and second objectives are transformed to minimize cases with multiplying by -1. Then, the problem is formulated by three objective functions and the group of constraints as follows:

$$\text{Min } f_1(x) = -x_1^2 - x_2^2 - x_3^2,$$

$$\text{Min } f_2(x) = -(x_1 - 1)^2 - x_2^2 - (x_3 - 2)^2,$$

$$\text{Min } f_3(x) = 2x_1 + x_2^2 + x_3,$$

Subject to:

$$g_1(x) = -x_1 + 3x_2 - 4x_3 + 6 \geq 0,$$

$$g_2(x) = -2x_1^2 - 3x_2 - x_3 + 10 \geq 0,$$

$$x \in R^3,\ 0 \leq x_1 \leq 3,\ 0 \leq x_2 \leq 4,\ 0 \leq x_3 \leq 2.$$

It is clear from *Table 3* that the initial solution is x_1= 0.5, x_2= 1.0, and x_3= $\sqrt{0.5}$ with distance = 16.3886. The first updated solution is: x_1= 0.25, x_2= 2.0, and x_3= 0.3536 with the distance =12.2311. The values of a_1 are 0.5, 2.0, and 0.5 for x_1, x_2, and x_3, respectively. But, the value of a_2 is zero for three variables. The best solution is: x_1^*= x_3^*= 0.0, x_2^*= 2.721654, and its distance = 9.072. This result is more accurate than the others.

Table 3. *The best point of example (3) by Corona approach.*

Data	Initial solution	Update solution		
		The first iteration	The second iteration	The third iteration
x_1	0.5	0.5 × 0.5 + 0.0 = 0.25	1.0 × 0.25 - 0.20 = 0.05	0.0 × 0.05 + 0.0 = 0.0
x_2	1.0	2.0 × 1.0 + 0.0 = 2.0	1.0 × 2.0 + 0.66933 = 2.66933	1.0 × 2.66933+ 0.052324 = 2.721654
x_3	$\sqrt{0.5}$	0.5 × $\sqrt{0.5}$ + 0.0 = 0.3536	0.0 × 0. 3536 + 0.0 = 0.0	0.0
C_1	5.7	10.33579	13.95799	14.164962
C_2	5.8	3.52145	1.98701	1.835038
f_1	-1.74	-4.1875	-7.12782	-7.4074005
f_2	-2.94	-7.27329	-12.02782	-12.4074005
f_3	2.7	4.85355	7.22532	7.4074005
f_1+ f_2 + f_3	-1.98	-6.60723	-11.93032	-12.4074005
D	16.388583	12.23112	9.20571	9.072179

Example 4:

This example has been introduced by Anov [23]. It consists of two nonlinear objectives, six decision variables with their bounds and six constraints as follows:

$$\text{Min: } f_1(x) = -(25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2),$$

$$\text{Min: } f_2(x) = (x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2),$$

Subject to:

$$x_1 + x_2 - 2 \geq 0,$$

$$-x_1 - x_2 + 6 \geq 0,$$

$$x_1 - x_2 + 2 \geq 0,$$

$$-x_1 + 3x_2 + 2 \geq 0,$$

$$-(x_3 - 3)^2 - x_4 + 4 \geq 0,$$

$$(x_5 - 3)^2 + x_6 - 4 \geq 0,$$

$$x_1, x_2, x_6 \in [0: 10], x_4 \in [0: 6], x_3, x_5 \in [1: 5].$$

When the proposed approach is applied to this example, the best solution is given as: $x_1^* = 5.0$, $x_2^* = 1.0$, $x_3^* = 1.0$, $x_4^* = 0.0$, $x_5^* = 1.0$, $x_6^* = 0.0$, $f_1^* = -242.0$, $f_2^* = 28.0$, $C_1 = 4.0$, $C_2 = C_4 = C_5 = C_6 = 0.0$, $C_3 = 6.0$, and its distance from the utopia point = 40.0. It is clear that the current approach doesn't require a significant computational effort to yield an accurate best solution with rapid convergence to the utopia point in the efficient set. But, Anov presented a group of efficient solutions to allow the decision-maker to select any possible best compromise solution.

Example 5:

This problem deals with the minimization of two nonlinear conflicting objective functions with two decision variables, two constraints, and two bounds for each variable.

$$\text{Min: } f_1(x) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2,$$

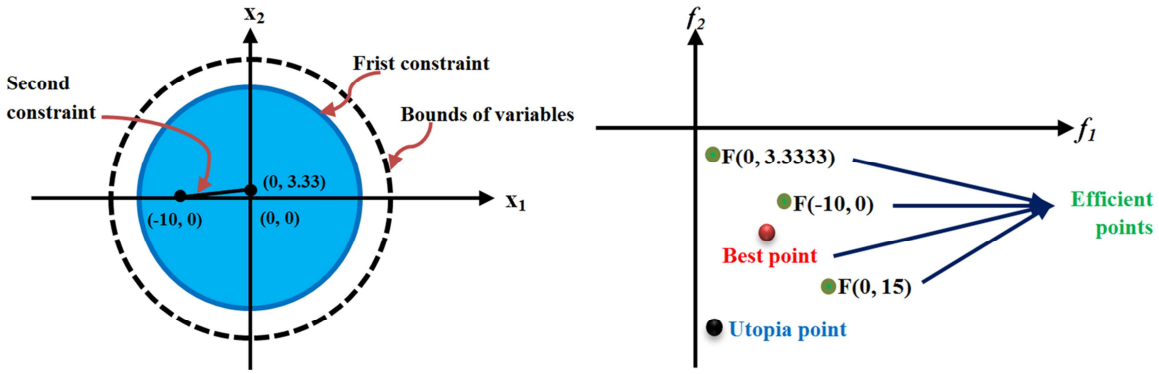
$$\text{Min: } f_2(x) = 9x_1 - (x_2 - 1)^2,$$

Subject to:

$$x_1^2 + x_2^2 - 225 \leq 0,$$

$$x_1 - 3x_2 + 10 \leq 0,$$

$$-20 \leq x_1 \leq 20, -20 \leq x_2 \leq 20.$$



(a) The decision space of example (5). (b) Some solutions in the objective space of example (5).

Figure 9. *The efficient set in the decision space and some solutions in the objective space of example (5).*

This example was solved by Chiandussi *et al.* [24] using four methods: The multi-objective genetic algorithm (MOGA), global criterion method, ε-constraint method, and the linear combination of weights method. The Pareto front obtained from the MOGA is linear with a very large number of iterations. The global criterion and the ε-constraint methods allow identifying many optimal solutions belonging to the Pareto front with a limited number of iterations. The linear combination method cannot choose the possible best compromise solution correctly. But, the best solution obtained by the Corona approach is: $x_1^* = -2.5$, $x_2^* = 10.5679$, $f_1^* = 113.7947$, $f_2^* = -114.0447$, $C_1 = -107.0695$, $C_2 = -24.2037$, and the distance from utopia point (D)= 146.646. The utopia point is: $f_1^* = 10.1$, $f_2^* = -217.739$. The Corona algorithm's run has lower number of steps than these previous methods. *Figure 9* shows the efficient set in the decision space and some solutions in the objective space.

Example 6:

This example is solved by Gebreel *et al.* [25] using the bacterial foraging and harmony search algorithms. Also, it is taken from the work of Abbas & Huda [26]. The original mathematical problem is formulated as follows:

$$\text{Min: } f_1(x) = x_1 x_5,$$

$$\text{Min: } f_2(x) = x_1^{-1} x_3^2 x_4^4,$$

Subject to:

$$5x_1^{-1} x_2 \leq 1,$$

$$2.5x_2^{-1} x_3^2 + 1.5x_3^{-1} x_4^{-0.5} x_5^{-0.5} \leq 1,$$

$x_i \geq 0$, where i = 1, 2, 3, 4, 5.

By applying the Corona model to this problem, the best solution is: $x_1^* = 8.76017$, $x_2^* = 1.752034$, $x_3^* = 0.4371853$, $x_4^* = 5.370617$, $x_5^* = 4.144123$, $f_1^* = 36.3032$, $f_2^* = 18.1516$, $C_1 = C_2 = 1.0$, and D = 40.588238. The corresponding optimum value is 54.4548, which is greater than the optimum value of the original problem (52.12682) without any weights in objective functions. But, the compromise optimal solution obtained by bacterial foraging and harmony search algorithms is: $x_1 = 1.271$, $x_2 = 0.2542$, $x_3 = 0.1665$, $x_4 = 5.3706$, and $x_5 = 28.5651$. The corresponding optimum value is equal to 54.45196, $C_1 =$

1.0000, and C_2= 0.999999314. Its distance from the utopia point is 40.5883. This means that the CV approach gives more robust result than others.

5. Conclusion

In the paper, an artificial Corona algorithm is presented as a new efficient tool for solving multi-objective programming problems. It simulates the epidemic behavior of the Corona virus that spreads throughout the world and infects healthy people. The inspiration for this model derives from the idea of virus transmission and treatment. This approach starts with one of three real numbers $(0.5, \sqrt{0.5}, 1.0)$ for each decision variable of a multi-objective problem. After that, it uses its linear formula to track the changing distance with iteration to determine an efficient solution with the shortest distance to the ideal solution. It is called the best efficient solution.

In the step of updating the solution, the parameters "a_1" and "a_2" in the Corona formula play an important role in setting the values of the variables. In each iteration of the algorithm, it is important to look at the values of objective functions and their distance from the utopia point with the feasibility of constraint(s). Moreover, the Corona approach can solve both linear and nonlinear problems without scalarization of the objectives at an acceptable computation time.

To illustrate the quality of a generated best solution by the proposed approach, some experimental examples are given. In some of these examples, the Corona approach gives explicitly better results compared to the other works of literature and also shows completely satisfactory results in other references. These examples are solved using Excel software to simplify the search process with accurate results.

Future Work

In future studies, this approach can be implemented in many different applications for a wide variety of domains.

Acknowledgements

Thanks are to ALLAH for his guidance and supports in showing us the path.

References

[1] Alia Youssef Gebreel (2021). Artificial Corona-inspired optimization algorithm: Theoretical foundations, analysis, and applications. *(Research gate), American Journal of Artificial Intelligence*, DOI: 10.11648/j.ajai.20210502.12, 5 (2), PP. 56-65.

[2] Kaisa M Miettinen (2004). Nonlinear multiobjective optimization. Kluwer Academic Publishers, Fourth Printing.

[3] Alia Youssef Gebreel (2021). Solving the multi-objective convex programming problems to get the best compromise solution. *(Research gate), Australian Journal of Basic and Applied Sciences*, DOI: 10.22587/ajbas.2021.15.5.3, 15 (5), PP. 17-29.

[4] Ashis Kumar Mishra, Yogomaya Mohapatra, & Anil Kumar Mishra (2013). Multi-objective genetic algorithm: A comprehensive survey. *International Journal of Emerging Technology and Advanced Engineering*, Website: www.ijetae.com, 3 (2), PP. 81-90.

[5] Ivan P. Stanimirovi´c, Milan Lj. Zlatanovi´c, & Marko D. Petkovi (2011). On the linear weighted sum method for multi-objective optimization, *FACTA* University *(NIS) SER. Math. Inform*., 26, PP. 49-63.

[6] Jason Brownlee (2012). Clever algorithms nature-inspired programming recipes, ISBN: 978-1-4467-8506-5.

[7] Giagkiozis, & P. J. Fleming (2015). Methods for multi-objective optimization: An analysis, *Information Sciences*, *Science Direct*, 293, PP. 1-16.

[8] Nyoman Gunantara (2018). A review of multi-objective optimization: Methods and its applications, *Cogent Engineering*, Taylor & Francis, DOI.org/10.1080/23311916. 2018.1502242, (5), PP. 1-16.

[9] Yin-Fu Huang, & Sih-Hao Chen (2020). Solving multi-objective optimization problems using self-adaptive harmony search algorithms. *Soft Computing*, *Springer-Verlag GmbH Germany*, DOI.org/10.1007/s00500-019-04175-0, 24, PP. 4081-4107.

[10] P. K. De, & Bharti Yadav (2011). An algorithm for obtaining optimal compromise solution of a multi objective fuzzy linear programming problem. *International Journal of Computer Applications*, (0975 – 8887), 17 (1), PP. 20-24.

[11] Neha Gupta, Irfan Ali, & Abdul Bari (2013). A compromise solution for multi-objective chance constraint capacitated transportation problem. *ProbStat Forum*, ISSN 0974-3235, 6, PP. 60-67.

[12] M. Upmanyu, & R. R. Saxena (2015). Obtaining a compromise solution of a multi objective fixed charge problem in a fuzzy environment. *International Journal of Pure and Applied Mathematics*, DOI: http://dx.doi.org/10.12732/ijpam.v98i2.3, 98 (2), PP. 193-210.

[13] Zhiyuan Wang, & Gade Pandu Rangaiah (2017) Application and analysis of methods for selecting an optimal solution from the Pareto-optimal front obtained by multiobjective optimization. *I&EC research*, *Industrial & Engineering Chemistry Research*, DOI: 10.1021/acs.iecr.6b03453, 56, PP. 560−574.

[14] Alia Youssef Gebreel (2016). On a compromise solution for solving multiobjective convex programming problems. *(Research gate), International Journal of Scientific & Engineering Research*, 7 (6), PP. 403-409.

[15] Alia Youssef Gebreel (2016). An adaptive interactive multi-objective optimization approach based on decision neural network. *(Research gate), International Journal of Scientific & Engineering Research*, 7 (8), PP. 1178-1185.

[16] Alia Youssef Gebreel (2018). Developing an Intelligent Interactive Approach for Multi-objective Optimization Problems. *(Research gate), International Journal of Scientific & Engineering Research*, http://www.ijser.or, 9 (12), PP. 1952- 1966.

[17] Murshid Kamal, Syed Aqib Jalil, Syed Mohd Muneeb, & Irfan Ali (2018). A distance based method for solving multiobjective optimization problems. *Journal of applied modern statistical methods*, DOI: 10.22237/jmasm/1532525455, Vol. 7, No. 1, PP: 1-23.

[18] Janett Walters-Williams, & Yan Li (2010). Comparative study of distance functions for nearest neighbors. Advanced Techniques in Computing Sciences and Software Engineering, Springer Science + Business Media B. V.

[19] Stephen Boyd, & Lieven Vandenberghet (2009). Convex optimization. Cambridge University Press, Second edition.

[20] Indresh Kumar Gupta, & Jeetendra Kumar (2015). VEGA and MOGA an approach to multi-objective optimization. *International Journal of Advanced Research in Computer Science and Software Engineering*, ISSN: 2277 128X, 5 (4), PP. 865-870.

[21] Sebastian Sitarz (2012). Interactive compromise hypersphere method and its applications. *RAIRO Operations Research*, DOI: 10.1051/ro/2012017, 46, PP. 235–252.

[22] Majid Rafei, Samin Ebrahim Sorkhabi, & Mohammad Reza Mosavi (2014). Multi-objective optimization by means of multi-dimensional MLP neural networks. *Neural Network World*, DOI: 10.14311/NNW.2014.24.002, (1/14), PP. 31- 56.

[23] Valdimir Sevasty Anov (2013). Hybrid multi- gradient explorer algorithm for global multi-objective optimization. *American Institute of Aeronautics and Astronautics*, eartius, Inc., 10 (94-99), PP. 1-15.

[24] G. Chiandussi, M. Codegone, S. Ferrero, & F. E. Varesio (2012). Comparison of multi-objective optimization methodologies for engineering applications. *Computers and Mathematics with Applications*, 63, PP. 912-942.

[25] Mohamed S. A. Osman, Waiel Fathi Abd El-Wahed, Mahmoud Mostafa El-Sherbiny, & Alia Youssef Gebreel (2018). Developing intelligent interactive approach for multi-objective optimization problems. Ph. D. in Operations Research Department, ISSR, Cairo University, Egypt, *(Research gate)*.

[26] Abbas Y. Al-Bayati, & Huda E. Khalid (2012). On multi-objective geometric programming problems with a negative degree of difficulty. *Iraqi Journal of Statistical Science*, 21, PP. 1-14.