

## Analyzing the performances of evolutionary multi-objective optimizers on design optimization of robot gripper configurations

Murat DÖRTERLER<sup>1</sup>, Ümit ATİLA<sup>2,\*</sup>, Rafet DURGUT<sup>2</sup>, İsmail ŞAHİN<sup>3</sup>

<sup>1</sup>Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara, Turkey

<sup>2</sup>Department of Computer Engineering, Faculty of Engineering, Karabük University, Karabük, Turkey

<sup>3</sup>Department of Industrial Design Engineering, Faculty of Technology, Gazi University, Ankara, Turkey

Received: 24.03.2020

Accepted/Published Online: 31.09.2020

Final Version: 27.01.2021

**Abstract:** Robot grippers are widely used in a variety of areas requiring automation, precision, and safety. The performance of the grippers is directly associated with their design. In this study, four different multiobjective metaheuristic algorithms including particle swarm optimization (MOPSO), artificial algae algorithm (MOAAA), grey wolf optimizer (MOGWO) and nondominated sorting genetic algorithm (NSGA-II) were applied to two different configurations of highly nonlinear and multimodal robot gripper design problem including two objective functions and a certain number of constraints. The first objective is to minimize the difference between minimum and maximum forces for the assumed range in which the gripper ends are displaced. The second objective is force transmission rate that is the ratio of the actuator force to the minimum holding force obtained at the gripper ends. The performance of the optimizers was examined separately for each configuration by using pareto-front curves and hyper-volume (HV) metric. Performances of the optimizers on the specific problem were compared with results of previously proposed algorithms under equal conditions. With respect to these comparisons, the best-known results of the configurations were obtained. Furthermore, the pareto optimal solutions are thoroughly examined to present the relationship between design variables and objective functions.

**Key words:** Robot grippers, engineering optimization, multiobjective optimization, design optimization, metaheuristic

### 1. Introduction

Robot grippers are mechanisms that are used in place of human hands in various jobs and processes in line with the requirements of efficiency, safety, sensitivity etc., which are again inspired and designed by human [1]. Robot grippers are widely used in production lanes, in environments threatening human life and health, in medical surgeries, and in the development of microelectromechanical systems [2]. The grippers, whose drive mechanism can be hydraulic, pneumatic, or electric, placed on the wrist part of the robot arm [3].

Although the function of the grippers is usually to grasp and carry an object without damaging it, they need to be designed and produced specifically for a job. Due to the complex structure of the grippers, their designs have been the subject of various optimization studies. Osyczka et al. for the first time, devised and formulated the design of grippers as a multiobjective continuous constraint optimization problem [4]. Osyczka [5] extended the problem by formulating the robot gripper design into five different gripper configurations. By applying multi objective genetic algorithm (MOGA) to the problem, he determined the optimal design parameters of these five configurations.

\*Correspondence: [umitatila@karabuk.edu.tr](mailto:umitatila@karabuk.edu.tr)

These configurations were generally accepted among the researchers and formed the basis of various studies in this field [6–10]. In these studies, various nature-inspired metaheuristic multiobjective optimization methods were used to solve the problem. Saravanan et al. [6] applied MOGA, nondominated sorting genetic algorithm II (NSGA-II) and multiobjective differential evolution (MODE) to optimize the gripper configurations described by Osyczka. Later, Datta and Deb [7] optimized the gripper configurations described by Osyczka using NSGA-II and obtained better results than Osyczka. However, when these conducted studies are examined, it is seen that some of the researchers did not obey the original configurations and made various modifications on them. Furthermore, there are inconsistencies in the application of the optimizers to the problem and the evaluation of their performance. Moreover, the computing resources used by these studies, which spread over the last 3 decades, differ as well. This situation does not allow a comparison on the performance of the used methods and the results of the studies.

In this study, we deal with the optimization of two robot gripper configurations originally described by Osyczka [5]. Beside NSGA-II, which was also applied by Saravanan [6] and Datta and Deb [7] to the same configurations before, we applied multi objective particle swarm optimization (MOPSO) and new generation metaheuristics such as multi objective grey wolf optimizer (MOGWO) and multi objective artificial algae algorithm (MOAAA) remaining loyal to the objectives of robot gripper problems described by Osyczka with equal conditions. MOPSO, MOGWO and MOAAA were applied to the mentioned gripper configurations for the first time in this study. As far as we know, the pareto fronts obtained by the MOAAA are the best ever achieved in the literature and there is no other study in the literature where these gripper configurations are optimized with current algorithms and analyzed with current performance metrics such as hyper volume.

The rest of this study is organized as follows. Section 2 presents the related studies on design optimization of robot grippers. Section 3 outlines the optimization methods applied to the problem. Experimental results and discussions were presented in Section 4. The paper is concluded in Section 5 with directions for future studies.

## 2. A brief literature on optimization of robot grippers

The first study on the optimization of robotic grippers was carried out by Cutkosky [11] on the selection, modeling, and design of the grasp on the grippers. In his study, Cutkosky developed an expert system to solve grasping problems. Bicchi and Kumar [12] studied on robotic-grasping and discussed the problems encountered on the design and control of robot grippers.

Osyczka et al. [4] considered gripper design optimization as a multiobjective problem and proposed a new GA-based optimizer to solve the problem. Osyczka [5] described the mathematical models of five different gripper configurations. Optimal design values are obtained by applying MOGA to each configuration.

Several studies have been conducted based on Osyczka's configurations. Rao et al. [8] simplified the problem of robot gripper design as a single-objective optimization problem and solved it with evolutionary algorithms. Saravanan et al. [6] applied three optimization algorithms (MOGA, NSGA-II, and multi objective differential evolution) to three different robot gripper configurations. Datta and Deb [7] applied NSGA-II to two different conservative configurations of Osyczka and achieved more optimal results than Osyczka [5] obtained. Furthermore, pareto-optimal solutions were extensively studied to establish a meaningful relationship between objective functions and variable values. Datta et al. [9] discussed the limitation of the lack of actuator analysis in Osyczka's problem and the original problem was modified. In this modified version, an actuator was assumed where the force was proportional to the actuator rigidity and applied voltage. Avder et al. [10] applied the

strength pareto evolutionary algorithm II (SPEA-II) to the problem.

There are different studies in the literature that do not rely on Osyczka's model. Lanni and Caccarelli [13] proposed a two-finger robot gripper mechanism for the multiobjective optimum design model, efficiency, acceleration, velocity, and four objective functions that were different in size were used. Li et al. [14] studied the three-grip mechanism for heavy-forging robot grippers and optimized link lengths and link angles by using multiobjective GA. The most important result of the study is the realization of small and large scale gripper designs with the best known dimensional values. The study of Dao and Huang [15] involved the multiobjective optimization of a gripper with two flexible elements to grasp objects. These flexible elements are the torsion spring torque and tension of the torsion spring. In the study, where Fuzzy-Taguchi method was used, the best-known values for the gripper mechanism horizontal and vertical force parameter values were found. Pahwa et al. [16] studied the optimization of geometric parameters to investigate the effect of dimensional change on robot gripper performance. In this study, it was determined that the performance of electro-thermal micro-grippers would be pretty much affected by dimensional changes.

In Zitar's [17] single-objective ant colony (ACO) algorithm, however, the optimization objective was to achieve a minimum holding force; the researchers determined the optimum finger force values to grasp hard objects, and it was also discovered that the most appropriate set of parameters for ACO's applicability to the problem was independent of the number of ants in each location. Ciocarlie et al. [18] studied the optimal design of a robotic gripper that performed fingertip and envelope grasping. In their study, the parameters of the course of active tendon and springs providing passive extension forces were optimized. As a result of the study, they achieved the best-known design with the combination of the dimensions of the optimized tendons and the activation parameters.

## 2.1. Osyczka's gripper configurations

Osyczka's [5] configurations contain two conflicting objective functions. One of them is the difference between the minimum and maximum forces (Eq. 18), and the other is the ratio of force transmission (Eq. 19). The aim of the optimization problem is to obtain the optimal link lengths of the robot gripper and the joint angles of the links by taking geometry and force constraints into account.

### 2.1.1. Configuration design of Gripper I

The schematic of Osyczka's Gripper I configuration is given in Figure 1. The configuration consists of seven design variables and eight nonlinear constraints. Six design variables (Eq. 1) consisting of joint lengths and joint angles with the range of values are shown by the (Eq. 2–8).

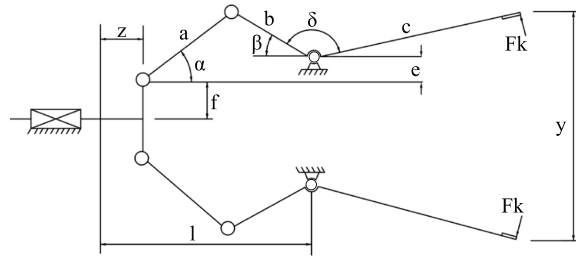
$$x = (a, b, c, e, f, l, \delta)^T \quad (1)$$

$$10 \leq a \leq 250 \quad (2)$$

$$10 \leq b \leq 250 \quad (3)$$

$$100 \leq c \leq 300 \quad (4)$$

$$0 \leq e \leq 50 \quad (5)$$



**Figure 1.** Gripper I mechanism scheme.

$$10 \leq f \leq 250 \tag{6}$$

$$100 \leq l \leq 300 \tag{7}$$

$$1.0 \leq \delta \leq 3.14 \tag{8}$$

where, design vector is shown as  $x = (a, b, c, e, f, l, \delta)^T$ . Here; while  $a, b, c, e, f$ , and  $l$  denote link lengths,  $\delta$  denotes the angle between links  $b$  and  $c$  [5].

Geometrical dependencies of Gripper I are given in Eq. 9–14 (Figure ??).

$$g = \sqrt{(l - z)^2 + e^2} \tag{9}$$

$$b^2 = a^2 + g^2 - 2.a.g. \cos(\alpha - \varphi) \tag{10}$$

$$\alpha = \arccos\left(\frac{\alpha^2 + g^2 - b^2}{2.a.g}\right) + \varphi \tag{11}$$

$$a^2 = b^2 + g^2 - 2.b.g. \cos(\beta + \varphi) \tag{12}$$

$$\beta = \arccos\left(\frac{b^2 + g^2 - \alpha^2}{2.b.g}\right) - \varphi \tag{13}$$

$$\varphi = \arctan\left(\frac{e}{l - z}\right) \tag{14}$$

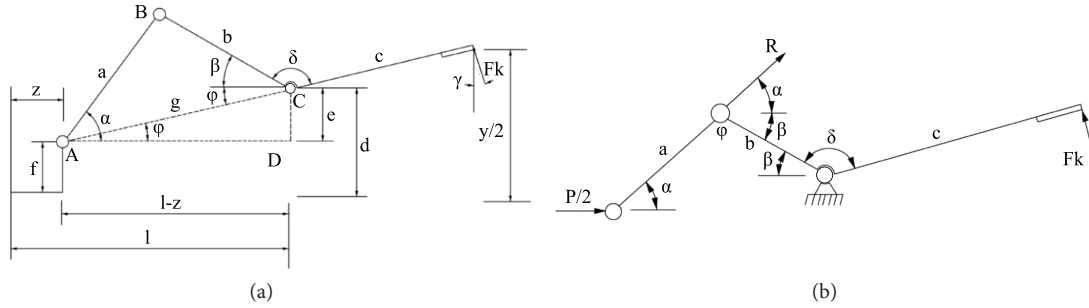
The free body diagram showing the force distributions of the Gripper I mechanism is shown in Figure ??. The equations for force distributions are shown in Eq. 15–17.

$$R \sin(\alpha + \beta) b = F_k c \tag{15}$$

$$R = \frac{P}{2. \cos \alpha} \tag{16}$$

$$F_k = \frac{P.b.\sin(\alpha + \beta)}{2.\cos\alpha} \quad (17)$$

where,  $R$  denotes the reaction force on the link  $a$ ,  $P$  indicates the force applied from the left side of the robot gripper mechanism for operation.



**Figure 2.** Gripper I mechanism (a) geometrical dependencies (b) force distribution.

The first objective function (Eq. 18) defined for Gripper I is to minimize the difference between minimum and maximum forces for the assumed range in which the gripper ends are displaced. Thus, the fluctuation in the holding force is minimized and the grip object is held in a fixed manner [5].

$$f_1(x) = \max_z F_k(x, z) - \min_z F_k(x, z) \quad (18)$$

The second important element for the gripper mechanism is the holding force obtained for the actuator force applied to the mechanism. The higher the minimum holding force, the more efficient the mechanism. Therefore, the second objective function given in Eq. 19 is the ratio of force transmission that is the ratio of the actuator force  $P$  to the minimum holding force obtained at the link  $c$  end. Minimizing this objective function ensures that the holding force obtained at the link  $c$  end is at the highest possible value.

$$f_2(x) = \frac{P}{\min_z.F_k(x, z)} \quad (19)$$

Both  $f_1$  and  $f_2$  depend on the distance  $z$ , which is a continuous value. Therefore, in order to find the values of  $f_1$  and  $f_2$  functions by using the given vector  $x$ ,  $z$  values between 0 and  $Z_{max}$ , which provide the minimum  $F_k(x, z)$  and maximum  $F_k(x, z)$  values should be determined.

Using the geometry of the gripper problem, nonlinear constraints given below are derived [5]:

$$g_1(x) = Y_{min} - y(x, Z_{max}) \geq 0 \quad (20)$$

$$g_2(x) = y(x, Z_{max}) \geq 0 \quad (21)$$

$$g_3(x) = y(x, 0) - Y_{max} \geq 0 \quad (22)$$

$$g_4(x) = Y_G - y(x, 0) \geq 0 \quad (23)$$

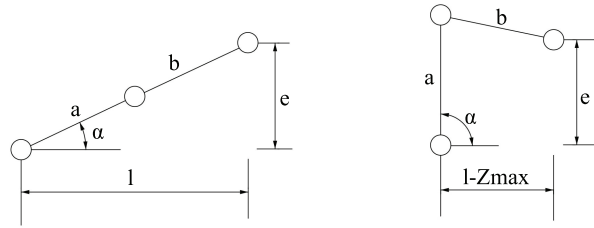
$$g_5(x) = (a + b)^2 - l^2 - e^2 \geq 0 \quad (24)$$

$$g_6(x) = (l - Z_{max})^2 - (a - e)^2 - b^2 \geq 0 \quad (25)$$

$$g_7(x) = l - Z_{max} \geq 0 \quad (26)$$

$$g_8(x) = \min F_k(x, z) - FG \geq 0 \quad (27)$$

where,  $g_1(x)$  constricts the fact that the distance between the gripping ends is smaller than the maximum size of the object being gripped for maximum displacement of the actuator;  $g_2(x)$  constricts the distance between the gripping ends corresponding to  $Z_{max}$ ;  $g_3(x)$  constricts the distance between the gripping ends that do not correspond to the actuator displacement;  $g_4(x)$  constricts the minimum displacement range of the gripping ends. With  $g_5(x)$  and  $g_6(x)$ , geometrical features of the gripper are maintained (Figure 3). Based on the geometry of the Gripper, the constraint  $g_7(x)$  is obtained. At the grippers, the minimum grasping force in terms of  $g_8(x)$  should be equal or greater than the limit of the grasping force selected.



**Figure 3.** Graphical representation of  $g_5(x)$  (left) and  $g_6(x)$  (right) constraints.

$y(x, z)$  shown in Eq. 20–23 expresses the distance between the gripper's grasping ends and is calculated through Eq. 28.  $Y_{min}$  and  $Y_{max}$  show the minimum and maximum sizes of the gripper mechanism, respectively.  $YG$  expresses the maximum distance between the gripper's grasping ends;  $Z_{max}$  expresses maximum distance of the actuator of the gripper;  $FG$  expresses the minimum default gripper force.

$$y(x, z) = 2[e + f + c \sin(\beta + \delta)] \quad (28)$$

Geometric parameters of the gripper mechanism are given below:

$$Y_{min} = 50mm, Y_{max} = 100mm, YG = 150mm, Z_{max} = 50mm, P = 100N, FG = 50mm.$$

### 2.1.2. Configuration design of Gripper II

The design vector,  $x$ , of the Gripper II configuration whose diagram is given in Figure ?? consists of five variables (Eq. 29).

$$x = (a, b, c, d, l)^T \quad (29)$$

$$10 \leq a \leq 250 \quad (30)$$

$$10 \leq b \leq 250 \quad (31)$$

$$100 \leq c \leq 250 \quad (32)$$

$$100 \leq d \leq 250 \quad (33)$$

$$100 \leq l \leq 250 \quad (34)$$

where,  $a$ ,  $b$ ,  $c$ ,  $d$ , and  $l$  indicate link lengths with the range of values and are shown by the Eq. 30–34. Geometrical dependencies of the Gripper II mechanism are expressed in Eq. 35–38 (Figure ??).

$$g = \sqrt{(l + z)^2 + d^2} \quad (35)$$

$$\alpha = \arccos\left(\frac{\alpha^2 + g^2 - b^2}{2.\alpha.g}\right) + \varphi \quad (36)$$

$$\beta = \arccos\left(\frac{b^2 + g^2 - \alpha^2}{2.b.g}\right) - \varphi \quad (37)$$

$$\varphi = \arctan\left(\frac{d}{l + z}\right) \quad (38)$$

Dependencies between the forces of Gripper II are given in Eq. 39–41.

$$R.b \sin(\alpha + \beta) = F.(b + c) \quad (39)$$

$$R = \frac{P}{2.\cos \alpha} \quad (40)$$

$$F_k = \frac{P.b \sin(\alpha + \beta)}{2.(b + c).\cos \alpha} \quad (41)$$

Objective functions used for the mechanism of the Gripper II are the same as those used for Gripper I.  $g_1$ ,  $g_2$ ,  $g_3$ ,  $g_4$  defined for Gripper I in Eq. 20–23 are applicable for Gripper II, as well. Other constraints used for Gripper II are as follows [5]:

$$g_5(x) = c - Y_{max} - a \geq 0 \quad (42)$$

$$g_6(x) = (a + b)^2 - d^2 - (l + z)^2 \geq 0 \quad (43)$$

$$g_7(x) = \alpha \leq \frac{\pi}{2} \quad \forall z \in (0, Z_{max}) \quad (44)$$

where,  $y(x, z)$  defined with Eq. 45 is the distance between the grasping ends of the gripper and  $Z_{max}$ ,  $Y_{min}$ ,  $Y_{max}$ ,  $YG$  are the same as ones given for Gripper I.

$$y(x, z) = 2 \cdot (d + (b + c)) \cdot \sin(\beta) \tag{45}$$

In solving of the Robot Gripper problem, minimum  $F_k(x, z)$  and maximum  $F_k(x, z)$  must be determined for possible different  $z$  values between 0 and  $Z_{max}$  with constraints provided. For this, the value of  $z$  between 0 and  $Z_{max}$  was increased by 0.5 step value and the minimum and maximum values of the objective function were determined. When the  $z$  value providing the constraints could not be determined, the existence of the current solution in the population was terminated with a high penalty score.

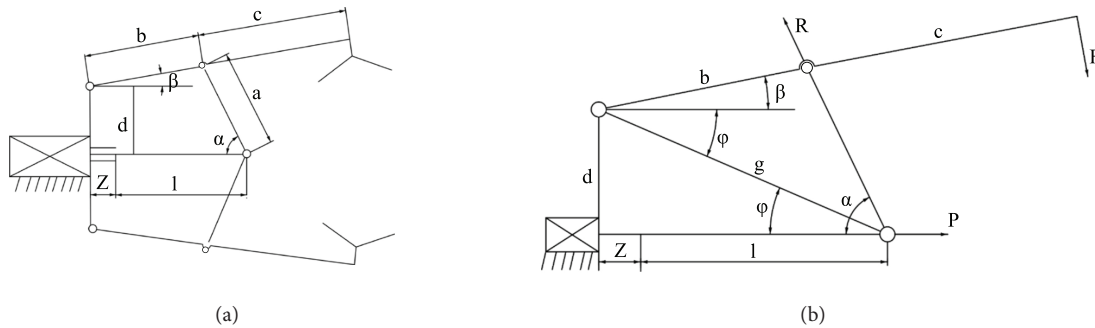


Figure 4. (a) Gripper II mechanism (b) Geometrical dependencies of Gripper II.

### 3. Optimization methods applied on the problem

The optimizers used in this study are multiobjective versions of popular metaheuristic algorithms. Genetic algorithm (GA), particle swarm optimization (PSO), grey wolf Optimizer (GWO) and artificial algae algorithm (AAA) are used to solve single objective optimization problems. These single-objective optimizers are developed into multiobjective versions by means of common additional methods.

#### 3.1. Genetic Algorithm

GA is an evolutionary search and optimization method developed by Holland in 1975 based on genetic and natural selection principles [19]. GA starts with the randomly creation of an initial population that consists of  $N$  number of solutions called chromosomes (Eq. 46). Chromosomes denoted by  $X_i = x_{i1}, x_{i2}, \dots, x_{id}$ , each containing one of the candidate solutions, are formed by the combination of significant genes [20].

$$x_{ij} = l_{ij} + r(u_{ij} - l_{ij}) \tag{46}$$

where  $i = 1, 2, \dots, N$ ,  $j = 1, 2, \dots, d$ ;  $d$  expresses the dimension of the numeric function,  $l_{ij}$  and  $u_{ij}$  express the upper and lower bounds of the dimension  $j$ ;  $r$  expresses a random uniformly distributed number between  $[0, 1]$ .

Traditionally, genetic processes are performed with crossover and mutation operators. The crossover operator combines the genetic materials of two chromosomes, called parents, to produce new individuals called offspring. Individuals with good characteristics that are likely to be selected are chosen to be parents of the



next generation and pass their genes to the next generation. The mutation alters one or more randomly selected genes of the chromosomes. Mutations add genetic diversity to the population [20].

The population is evolved towards the best solution by crossover and mutation operators throughout iterations (Figure 5). The first step of an iteration begins with the calculation of the fitness values of each chromosome in the population. Then, parents selected for reproduction are replaced by crossover and transformed with mutation to create offspring chromosomes. Parent chromosomes that are not passed through crossover or mutation and newly generated child chromosomes form a new population. This new population creation process is repeated for the specified number of times or until the time that no better chromosomes are generated [21].

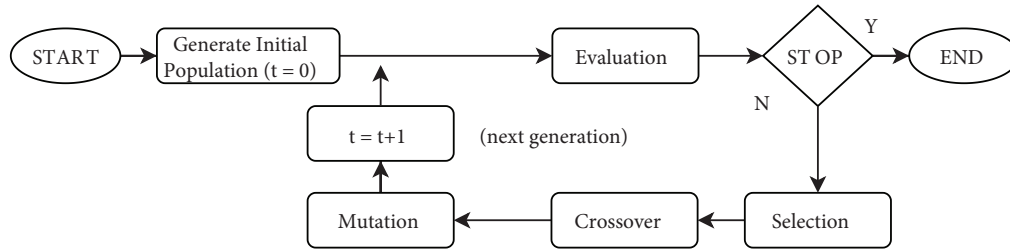


Figure 5. Steps of Genetic Algorithm

### 3.2. Particle swarm optimization

PSO, a swarm-based optimization algorithm, was proposed by Kennedy et al. in 1995 [22]. A global optimum solution is searched in PSO by considering the position and speed of each particle forming the swarm. In the first stage of the method, the position  $X_i = x_{i1}, x_{i2}, \dots, x_{id}$  and the velocity vector  $V_i = v_{i1}, v_{i2}, \dots, v_{id}$  are randomly generated in accordance with the method in Eq. (46) to form the initial population for N number of solutions. Then the position  $X_i'$  and vector  $V_i'$  values of each particle in the next iteration are updated as per Eqs (47, 48).

$$V_i' = wV_i + c_1r_1(pbest_i - X_i) + c_2r_2(gbest - X_i) \quad (47)$$

$$X_i' = X_i + V_i' \quad (48)$$

where,  $pbest_i$  expresses the best position that the particle i has ever achieved;  $gbest$ , the best position the swarm has ever achieved;  $c_1$  and  $c_2$  express the two constants representing acceleration coefficients, which generally take the value of 2.0;  $r_1$  and  $r_2$  express a number randomly distributed between [0, 1]; and  $w$  expresses the linearly decreasing inertia weight in the range of [0.9, 0.4] during iterations. The algorithm repeats this cycle until it reaches the predetermined criteria of dismissal.

### 3.3. Grey wolf optimizer

GWO, proposed by Mirjalili et al. in 2014 is a metaheuristic optimization method inspired by the behavior of the grey wolves in nature and the hierarchical structure within the flock [23]. The wolves in the flock can have one of four different roles. The first three fittest individuals are defined as  $\alpha, \beta$ , and  $\delta$ , respectively. These three individuals lead the other individuals in the flock defined as  $w$ .

In the first stage of the method, the position  $X_i = x_{i1}, x_{i2}, x_{id}$  is randomly generated in accordance with the method in Eq. (46) to form the initial population for  $N$  number of solutions. The next position of each wolf is determined by Eq.s (49,50).

$$D_i = |CX_p - X_i| \quad (49)$$

$$X'_i = |X_p - AD_i| \quad (50)$$

In the Eq. (49),  $X_p$  is the position vector of the prey,  $X_i$  is the position of the  $i^{th}$  wolf in the flock,  $D_i$  is the distance between  $X_p$  and  $X_i$ .  $A$  and  $C$  are the coefficient factors and calculated with the help of Eq.s (51, 52):

$$A = 2ar_1 - a \quad (51)$$

$$C = 2r_2 \quad (52)$$

While  $a$  decreases linearly in the range of  $[2.0, 0.0]$  during iterations;  $r_1$  and  $r_2$  are random values between  $[0, 1]$ . The next positions of the solutions are calculated through Eq.s (53-55) with respect to the positions of  $\alpha, \beta$ , and  $\delta$ , the fittest three in the pack, since the position of the prey is not known.

$$D_\alpha = C_1X_\alpha - X_i, \quad D_\beta = C_2X_\beta - X_i, \quad D_\sigma = C_3X_\sigma - X_i \quad (53)$$

$$X_1 = X_\alpha - A_1D_\alpha, \quad X_2 = X_\beta - A_2D_\beta, \quad X_3 = X_\sigma - A_3D_\sigma, \quad (54)$$

$$X'_i = \frac{X_1 + X_2 + X_3}{3} \quad (55)$$

The algorithm repeats this cycle until it reaches the predetermined criteria of dismissal.

### 3.4. Artificial algae algorithm

AAA is a swarm-based heuristic optimization algorithm, which is inspired by the behavior of algae in accessing the substances they need to produce food [24]. In this method, a possible solution is represented by the position of an algal colony, and its fitness value is represented by the nutrient concentration of the colony.

Each iteration of AAA consists of three main phases: helical motion, evolutionary process, and adaptation. In the first stage of the method, the position  $X_i = x_{i1}, x_{i2}, x_{id}$  is randomly generated in accordance with the method in Eq. (46) to form the initial population for  $N$  number of solutions.  $X_i$  represents an algal colony, and  $x_{ij}$  represents an algal cell that belongs to that colony. The position of each colony in the next iteration ( $X'_i$ ) is then updated as per the Eq.s (56-61), where the helical motion is modeled.

$$x'_{im} = x_{im} + (x_{jm} - x_{im})(\Delta - \tau(x_i))p \quad (56)$$

$$x'_{ik} = x_{ik} + (x_{jk} - x_{ik})(\Delta - \tau(x_i))\cos\alpha \quad (57)$$

$$x'_{iz} = x_{iz} + (x_{jz} - x_{iz})(\Delta - \tau(x_i))\sin\alpha \quad (58)$$

$$\tau(x_i) = 2\pi\left(\sqrt[3]{\frac{3G_i}{4\pi}}\right) \quad (59)$$

$$G'_i = G_i + \mu_i G_i \quad (60)$$

$$\mu_i = \frac{\mu_{max} f(x_i)}{\frac{G_i}{2} + f(x_i)} \quad (61)$$

where  $x_{im}$ ,  $x_{ik}$ , and  $x_{iz}$  are three algae cells randomly selected from the  $i^{th}$  algal colony ( $m^{th}$ ,  $k^{th}$ , and  $z^{th}$  cells),  $x_{jm}$ ,  $x_{jk}$ ,  $x_{jz}$  are algae cells in selected sizes of a different algae cell in the population determined as the light source,  $\alpha, \beta \in [0, 2\pi]$ ;  $c, p \in [-1, 1]$ ,  $\Delta$  is the coefficient of cutting force.  $\tau(x_i)$ , however, is the friction coefficient of the  $i^{th}$  algal colony at time  $t$ ;  $G_i$  is the size of this colony,  $i$  is the growth rate of the colony,  $\mu_{max}$  is the maximum growth rate which is considered to be 1, and  $f(x_i)$  is the fitness value.

In the evolutionary process, in the place of a randomly selected cell of the smallest algal colony, a cell of a same size within the largest colony is copied. In this process, an algal colony, which has not been able to grow sufficiently in its environment, tries to resemble the largest colony in the same environment to survive during the adaptation stage. The algal colony with the highest starving level is subjected to adaptation after a helical motion cycle.

### 3.5. Adaptation of the single objective algorithms to the multi-objective ones

There is a set of solutions, called pareto solutions, in multiobjective optimization instead of a single optimal solution. All pareto solutions are equally important in terms of objective functions. Thus, it is difficult to define both the best and worst solutions and to compare two solutions. Therefore, additional modifications are required to solve multiobjective optimization problems with single objective optimizers. However, these modifications are made with a strategy that algorithms use when comparing the two solutions, not with the one in the algorithms' own solution update mechanisms. The pseudo-code of the method used for the conversion of a single objective optimizer into a multiobjective optimizer is given in Algorithm 1.

---

**Algorithm 1** Pseudo code for adopting metaheuristic algorithms to multiobjective

---

- 1: Initialize  $P_0$  randomly in  $S$  and evaluate  $F$  for  $P_0$
  - 2: Sort  $P_0$  based on nondominated sort method
  - 3: Find  $NDR$  and fronts
  - 4: Compute  $CRD$  for each front
  - 5: Get updated solutions ( $P_j$ ) using Metaheuristic Algorithm
  - 6: Merge  $P_0$  and  $P_j$  in  $P_i$  ( $P_i = P_0 \cup P_j$ )
  - 7: For  $P_i$  perform Step 2
  - 8: Sort  $P_i$  based on  $NDR$  and  $CRD$
  - 9: Replace  $P_0$  with first  $N_{pop}$  members of  $P_i$
- 

, where  $N_{pop}$  is the population size and  $S$  is the feasible solution set. An initial population  $P_0$  is created from  $S$  randomly. Objective function values are calculated for each solution in  $P_0$  and the values are assigned to the  $F$  vectors of the solutions. In the next step, the computations of nondominated sorting and crowding distance are performed on  $P_0$ . The single-objective version of the optimizer is performed on  $P_0$  to obtain new

population  $P_j$ .  $P_i$  is obtained by merged  $P_0$  and  $P_j$ . The solutions in  $P_i$  are ranked according to the values of non-dominated ranking (NDR) and crowding-distance (CRD).  $P_0$  is updated by replacing its solutions with the best  $N$  solutions in the ranked  $P_i$ . These processes are repeated until the maximum number of iterations is reached [25].

The values of NDR, calculating by fast nondominated sorting method, and CRD, calculating by crowding-distance methods, are used to compare between the existing and the current solutions. If the NDR value of the current solution is better than the NDR of the existing solution, current solution is considered as the better solution. If the NDR values of the solutions are equal, the solution with a better CRD value is considered as the better. The solution with the best NDR value and CRD values is considered the best solution in a population. Conversely, the worst solution is the one with the worst values of NDR and CRD [25, 26].

Each solution is compared to the other ones in the population according to the roles of pareto domination in Fast non-dominated sorting method. Non-dominated solutions form the first non-dominated front and NDR values of these solutions are equal to 1. The remaining non-dominated solutions in the population are selected to form the second front and NDR values of these solutions are equal to 2. Likewise, the remaining solutions are placed to the fronts according to their non-domination ranks. Thus, all solutions are placed to the fronts by sorting according to their nondomination levels [25, 26].

Crowding-distance is an approach to protect diversity in the population. CRD value of a solution is calculated with the help of two nearest neighbors of it and extreme solutions at the same degree of the solution. The solutions on the related front are ranked increasingly according to objective function values. CDR values of the first and the last solutions on the ordered front are set to infinity in order to protect the extreme solutions of each objective in the population. CRD values of the other solutions on the front are calculated by Eq. (62) [26].

$$CRD_i^j = \frac{objF_j^{i+1} - objF_j^{i-1}}{objF_j^N - objF_j^1} \quad (62)$$

, where  $j$  is the index of the objectives in the problem,  $i$  is the index of the solutions on the front,  $N$  is the total number of solutions. CRD values are calculated as the number of objective functions for each solution on the front. Ultimate CRD value of a solution on the front calculated by adding up the CRD values for the objectives of the solution [26].

#### 4. Results and Discussion

Robot gripper design optimization problem was solved with NSGA-II, MOGWO, MOAAA, and MOPSO. For comparison, the results of Osyczka [5] obtained with MOGA were directly taken from his study and described as “original” in our study. As Datta and Deb [7] did not include the numerical results of NSGA-II method in their own studies, in order to make a robust comparison, NSGA-II method was reapplied to the problem following the same parameters in the study. The performance of each optimizer was examined separately by using pareto-front curves and hypervolume metric (HV) [27,28]. In addition, the characteristics of the design variables were revealed through analysis of the obtained optimal solutions. The algorithms were run on a computer with Intel Core™ i7 and 2.6 GHz CPU and 8GB RAM, using MATLAB 2016b software (MathWorks, Inc., Natick, MA, USA) on 64-bit Windows 10 operating system.

#### 4.1. Experimental settings and parameter tuning

For fair performance comparison, the population size was selected as 100 for all algorithms. Number of function evaluation was used as stopping criterion and set to  $10e^4$ . All algorithms were run 30 times with random seeds. The front size value of the algorithms was selected as 200 for pareto-front analysis.

Besides these general conditions, metaheuristic algorithms have their own parameters. The performance of these algorithms varies according to the values selected for the parameters. For this reason, proper parameter value selection is necessary to reach optimal solutions.

Since evolutionary and adaptation processes of AAA reduce diversity and convergence, MOAAA uses polynomial mutation method instead. NSGA-II also uses polynomial mutation and for both MOAAA and NSGA-II, probability of simulated binary crossover (SBX) was selected as 0.9 and mutation probability was selected as 0.1. For real-variable SBX and real-variable mutation, distribution indexes were selected as 20 and 100, respectively.

For MOGWO, number of grids per each dimension was selected as 20 and gamma value was set as 2. For MOPSO, number of grids per each dimension was selected as 7, inertia weight was selected as 0.5, inertia weight dumping rate as 0.99, personal learning coefficient as 1, global learning coefficient as 2, gamma value as 2 and mutation rate as 0.1.

In addition, parameter tuning was performed to determine the appropriate values of the two parameters of MOAAA, MOGWO and MOPSO used in solving Gripper I and Gripper II problems. In this context, cutting force ( $\Delta$ ) and energy loss ( $e$ ) parameters were selected for MOAAA, grid inflation ( $\alpha$ ) and leader selection pressure ( $\beta$ ) were selected for MOGWO, and MOPSO. Each algorithm was run 30 times with the defined parameter values in the scheme and the average HV values were calculated and presented in Table 1 and Table 2. The most suitable parameter values found for the algorithms were summarized in Table 3.

#### 4.2. Results for Gripper I

The pareto optimal curves of solutions, obtained by the applied optimizers to the problem were presented in Figure 6. At the end of optimization, it was aimed to store 200 solutions, but number of solutions found by MOGWO and MOPSO was much lower than those found by other algorithms and was in range between 5 and 20. While the results obtained by MOAAA dominated the results obtained by NSGA-II, the results were slightly better than the solutions obtained by MOGA. In addition, the performance of MOPSO and MOGWO algorithms was lower compared to other algorithms.

The performance of the optimizers applied to the problem was also examined in terms of hypervolume (HV) indicator. Because the problem had two objectives, hypervolume indicator was computed as hyper area [28]. By running each algorithm 30 times, the HV values of the solutions obtained in pareto-front curves were calculated. Maximum f1, f2 points in the nondominant solution set of algorithms were selected as the reference point in the calculations. Statistical evaluation of the obtained results was given in Table 4.

NSGA-II achieved the best HV value, which was the maximum one of the obtained scores. However, in terms of minimum and mean HV values, MOAAA was more successful. Although MOGWO seemed more stable with a lower standard deviation than other optimizers, this was because MOGWO found very few results. Similarly, the number of solutions found by MOPSO was also much lower than those found by other algorithms. When we compare MOAAA and NSGA-II in terms of stability, MOAAA worked more stable with a small difference. Following results presented in this subsection were based on MOAAA since it obtained the best mean performance.

**Table 1.** HV values obtained with parameter scheme for Gripper I.

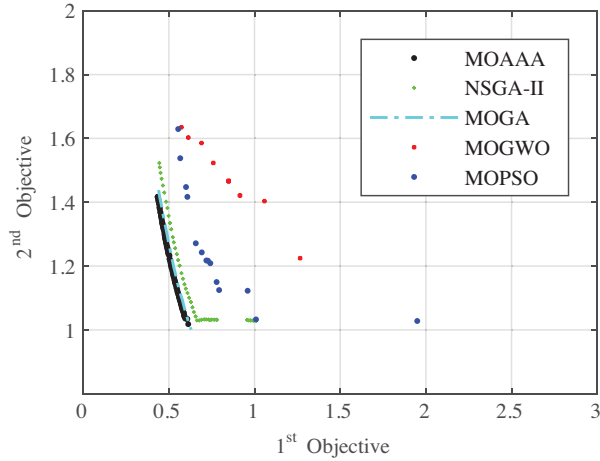
MOAAA					
$e / \Delta$	0.1	0.2	0.3	0.4	0.5
1	0.198864	0.188713	0.212175	0.458474	0.286629
2	0.736424	0.239787	0.162109	0.150434	0.150589
3	1.762455	0.188714	0.181339	0.231712	0.426979
4	0.199148	0.723451	0.348524	0.944471	0.904897
MOGWO					
$\alpha / \beta$	0.1	0.2	0.3	0.4	0.5
1	1.019515	0.721256	6.60703	1.417709	1.13278
2	1.470648	1.890653	1.070068	1.669965	3.420685
3	0.838014	3.901131	0.829132	5.596999	1.1427
4	2.803807	0.335991	0.79851	1.133351	1.688593
MOPSO					
$\alpha / \beta$	0.1	0.2	0.3	0.4	0.5
1	6.211303	0.555267	1.530974	0.565418	2.136324
2	0.470426	1.324686	1.764479	1.292824	1.961765
3	1.181807	1.523437	1.552609	0.65545	1.090087
4	3.139487	0.931006	3.769469	1.172927	0.973215

**Table 2.** HV values obtained with parameter scheme for Gripper II.

MOAAA					
$e / \Delta$	0.1	0.2	0.3	0.4	0.5
1	1721.235	1576.286	1666.635	1842.619	1508.368
2	1486.26	1490.54	1487.691	1495.513	1490.809
3	1476.103	1475.07	1463.23	1453.479	1457.599
4	1447.984	1442.858	1466.505	1467.927	1460.512
MOGWO					
$\alpha / \beta$	0.1	0.2	0.3	0.4	0.5
1	1348.758	795.0987	410.6486	377.0293	584.2573
2	1151.249	473.981	730.0036	1221.415	662.7884
3	422.7638	2355.98	640.6975	1161.351	433.8639
4	374.6938	942.4351	620.3936	826.2959	1775.361
MOPSO					
$\alpha / \beta$	0.1	0.2	0.3	0.4	0.5
1	2876.173	993.5168	1876.043	1202.197	895.4167
2	10969	1401.511	1738.941	6319.743	1211.569
3	41828.03	2370.034	1480.112	1033.176	2481.02
4	6471.683	1733.632	2727.489	1241.717	1122.926

**Table 3.** Best configurations for set of selected parameters

	Gripper I		Gripper II	
	$e/\alpha$	$\Delta/\beta$	$e/\alpha$	$\Delta/\beta$
MOAAA	0.1	3	0.4	1
MOPSO	0.1	1	0.1	3
MOGWO	0.3	1	0.1	3

**Figure 6.** Pareto optimal curves of the solutions.**Table 4.** Comparison of optimizers for Gripper I based on HV metric.

Optimizer	Minimum	Maximum	Mean	Std. Dev
MOPSO	0.412628	21.0656	6.556675	5.166969
MOGWO	0.173416	18.20551	5.560682	<b>4.380597</b>
MOAAA	<b>0.835446</b>	22.74119	<b>7.771376</b>	5.605863
NSGAI	0.563397	<b>23.1578</b>	7.712539	5.704334

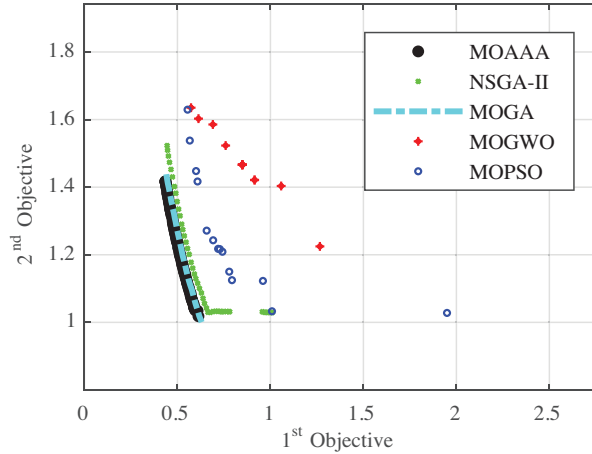
Five representative solutions from the pareto set of MOAAA for the Gripper I mechanism were presented in Table 5. The exemplary designs given here do not have any superiority over the others. They are all examples of optimal designs. As can be seen from Table 5, it is enough to adjust the design variables  $c$ ,  $f$  and  $\delta$  to obtain different optimal designs in Gripper I. Other parameters have similar values in all optimal designs. Figure 7 demonstrates how design variables change with respect to the second objective function (force transmission ratio -  $f_2$ ).

As it can be seen from the graphs, the variable  $a$  was 250, its upper limit value; the variables  $b$ ,  $e$  and  $l$  were fixed at 229, 13, 216, respectively. Design variables that made the difference between optimal designs were variables  $c$ ,  $f$  and  $\delta$ . Here, the variable  $c$  (gripper link length) changed as monotonous to the force transmission ratio ( $f_2$ ) as shown in Figure 7c and the relationship between them can be expressed linearly as  $c = 189.312 \times f_2$ .

The results obtained in the study showed that the length of link  $c$  should be larger for higher force

**Table 5.** Some nondominated solutions for Gripper I.

	$f_1(x)$	$f_2(x)$	$a$	$b$	$c$	$e$	$f$	$l$	$\delta$
1	0.467	1.48	250	229.2489139	288.8811795	13.17761212	32.07097718	216.2533262	1.968850801
2	0.544	1.271	250	229.2489139	248.0335479	13.16949941	25.72711831	216.2621674	1.943711437
3	0.559	1.236	250	229.2489139	241.2300885	13.17112009	29.52630403	216.2621674	1.963142144
4	0.655	1.0589	250	229.2489139	206.6813311	13.18285929	26.09867465	216.1365434	1.955990015
5	0.656	1.053	250	229.2489139	205.5625523	13.16749119	25.29606726	216.2685968	1.955461294

**Figure 7.** Variations of link length for Gripper I: (a)  $a$  with  $f_2$  (b)  $b$  with  $f_2$  (c)  $c$  with  $f_2$  (d)  $\delta$  with  $f_2$  (e)  $e$  with  $f_2$  (f)  $f$  with  $f_2$  (g)  $l$  with  $f_2$ .

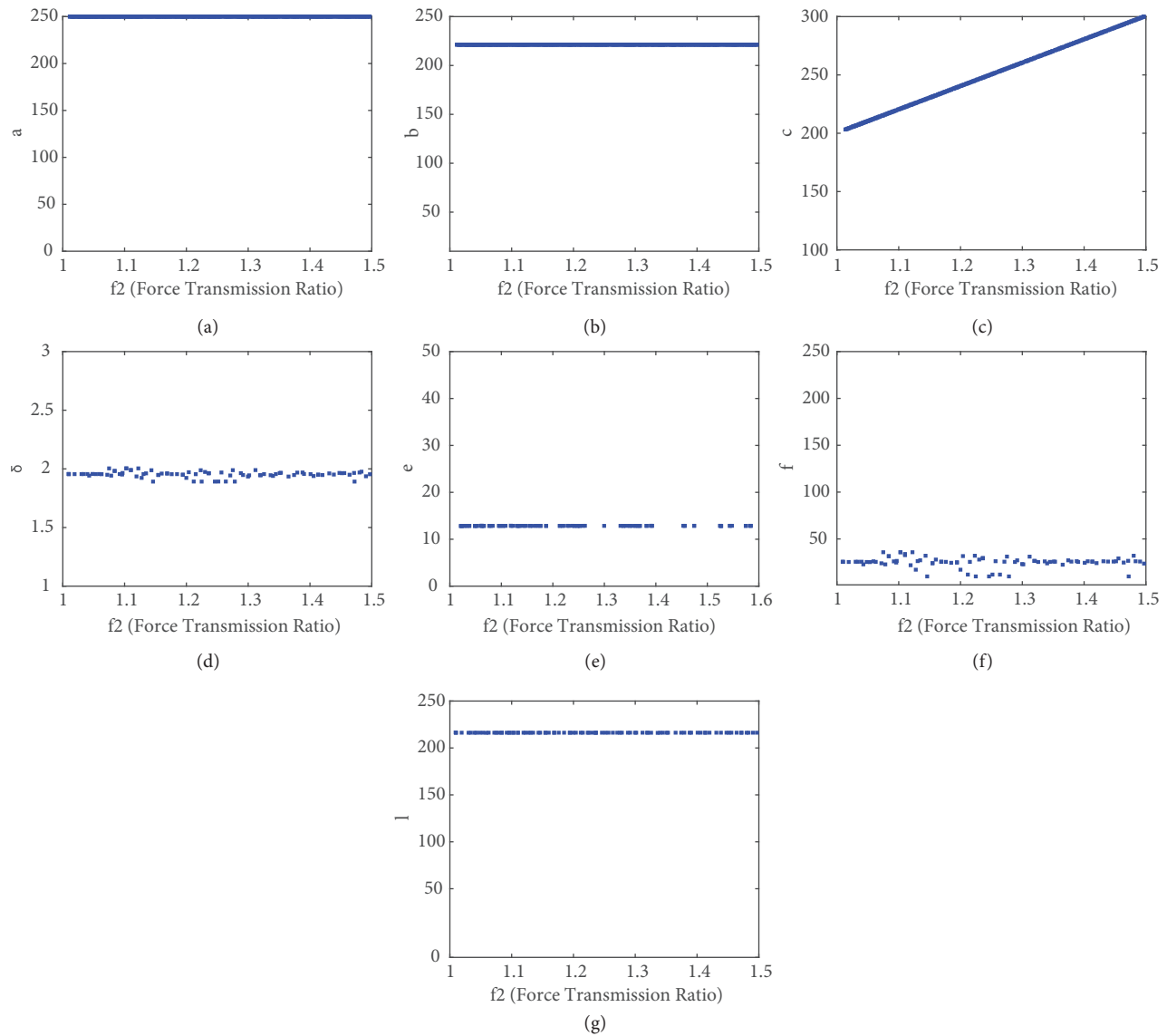
transmission ratio and smaller for lower force transmission ratio. Thus, selecting the link length of  $c$  high will cause  $f_2$  value to be high. In addition, while the other design variable,  $f$ , showed an irregular distribution between 10 and 35, the variable  $\delta$  varied between 1.89 and 2.04. There was no monotonous relationship between these two variables and  $f_2$ . The results showed that for a better gripper design, it was not enough to select only the link length of  $c$ . In addition, the appropriate values should also have been selected for the variables  $f$  and  $\delta$ . Datta and Deb [7] stated in his study that only the variable  $c$  had a monotonous relationship with  $f_2$ , and the other variables took constant values. However, it was observed that the results obtained by Datta and Deb for Gripper-I design did not meet the  $g_1(x)$  and  $g_3(x)$  constraints according to the values of the presented solutions. Consequently, the relationships between the design variables and  $f_2$  in [7] were not verified.

### 4.3. Results for Gripper II

The pareto optimal curves of algorithms obtained for Gripper II configuration were presented in Figure 8. At the end of optimization, it was aimed to store 200 solutions, but average number of solutions found by MOGWO and MOPSO was around 50, almost a quarter of other algorithms. As shown in Figure 8, the fronts obtained by MOAAA and NSGA-II were very close to each other and slightly dominated the results obtained by MOGWO and MOPSO. Finally, quality of solutions obtained by MOGA algorithm was far behind compared to other algorithms.

The performances of the optimizers calculated in terms of HV metric for Gripper II configuration were presented in Table 6. It was observed that MOAAA achieved the best score compared to other algorithms.





**Figure 8.** Pareto optimal curves of the solutions.

Moreover, MOAAA obtained better mean score than other algorithms. NSGA-II took the second place and was closest one to MOAAA in terms of mean score. In terms of standard deviation, it can be said that MOGWO and MOPSO worked slightly more stable than MOAAA and NSGA-II. Following results presented in this subsection were based on MOAAA since it obtained the best mean performance.

Five representative solutions from the pareto set of MOAAA for the Gripper II mechanism were presented in Table 7. As can be seen from Table 7, it is enough to adjust the design variables  $b$  and  $d$  to achieve different optimal designs in Gripper II. Other parameters have similar values in all optimal designs.

Figure 9 illustrates how the design variables change with respect to the second objective function (force transmission ratio).

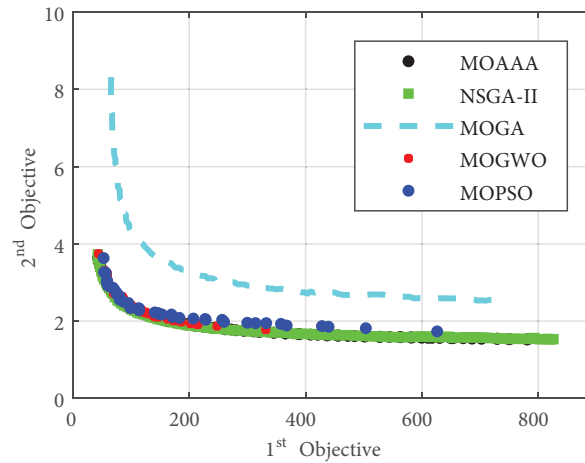
As it can be seen from the graphs, while the variable  $a$  was fixed at 150, variables  $c$  and  $l$  were fixed

**Table 6.** Comparison of optimizers for Gripper II based on HV metric

Optimizer	Minimum	Maximum	Mean	Std. Dev
MOPSO	1657.26	180128.69	16992.70	34415.12
MOGWO	1619.445	168466.91	16258.019	<b>32461.36</b>
MOAAA	1741.50	<b>202905.05</b>	<b>18494.86</b>	38379.54
NSGAII	<b>1745.85</b>	196981.69	18053.02	37305.26

**Table 7.** Some nondominated solutions for Gripper II.

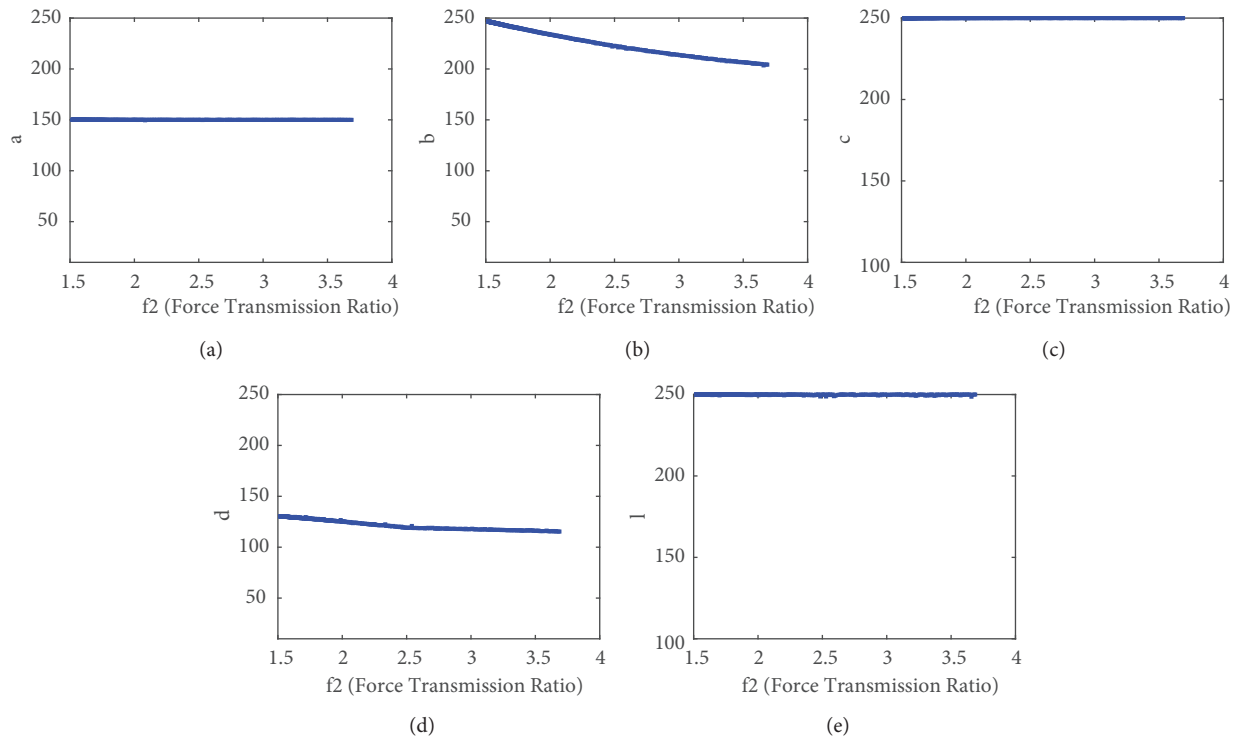
	$f_1(x)$	$f_2(x)$	$a$	$b$	$c$	$d$	$l$
1	47.681	3.380	149.865	207.448	249.948	116.379	249.078
2	94.532	2.367	149.974	224.698	250	121.221	249.042
3	455.748	1.623	149.915	243.646	249.949	129.460	249.948
4	653.587	1.557	149.572	245.280	249.659	129.460	249.619
5	749.612	1.532	149.974	246.224	250	130.197	249.995

**Figure 9.** Variations of link length for Gripper II: (a)  $a$  with  $f_2$  (b)  $b$  with  $f_2$  (c)  $c$  with  $f_2$  (d)  $d$  with  $f_2$  (e)  $l$  with  $f_2$ .

at 250, which was the upper limit in all solutions. Design variables that made the difference between optimal designs were the variables  $b$  and  $d$ . Here, the relationship between the link length  $b$  and force transmission ratio ( $f_2$ ) can be expressed as  $b = 282.789 \times f_2^{-0.298}$ . The relationship between the other link length  $d$  and force transmission ratio ( $f_2$ ) can be expressed as  $d = 142.631 \times f_2^{-0.184}$ . Although the design variables  $b$  and  $d$  in Gripper II varied within a small region of their variable range, they had the effect of creating a conflict between the two objective functions involved in the problem. With the increase in  $f_2$ , the variable value  $b$  decreased. This can be explained by the fact that the variables  $b$  and  $c$  behaved as an equilibrium in proportion to the end forces  $P$  and  $F_k$ . This is because  $f_2 = P/F_k$  and for a constant  $c$ ,  $f_2$  is inversely proportional to  $b$ .

#### 4.4. Comparison of gripper I and II

Choosing a better design in terms of the objective functions used for Gripper I and II designs is an important concern for end users. Even though the designs are similar, there are operational differences between them. Compared to Gripper I, Gripper II has less connections. Moreover, while Gripper-I closes tips when the actuator is moved to the right, the Gripper-II closes tips when it is moved to the left. Minimizing the first objective, which is the difference between maximum and minimum gripping forces, ensures that the gripping force does not change much over the entire operating range of the gripper. As seen in Figure 6 and Figure 8, the changes in gripping force  $F_k$  are higher than in Gripper II. In addition, Gripper I has a better force transmission ratio. Thus, when comparing the pareto-optimal solution curves of the gripper designs given in Figure 10, it can be said that Gripper I is generally a better design compared to Gripper II. As seen in Figure 10, Gripper I achieves better force transmission ratio compared to Gripper II.



**Figure 10.** Comparison of pareto-optimal solution curves of two gripper designs.

When Gripper I configuration is considered, it is seen that the problem has 7 design variables and 8 constraints. As can be seen in Fig 6, the MOAAA algorithm achieves the most successful results for this configuration, while the results are close to the MOGA algorithm. NSGA II achieves a slightly worse pareto front curve than these two methods. On the other hand, MOGWO and MOPSO algorithms achieves a limited number of pareto curves with solutions and stand behind other algorithms. The Gripper II configuration contains fewer decision variables and constraints than Gripper I. There are 5 decision variables and 7 constraints in this configuration. As can be seen in Fig 8, MOAAA and NSGA-II presents the best pareto curves in Gripper II configuration with slightly better solutions than MOGWO and MOPSO. Contrary to its performance in the Gripper I problem, the MOGA shows much lower performance in Gripper II problem and is far behind other

algorithms.

## 5. Conclusion

In this study, the optimization of two configurations of robot gripper design is discussed. Analysis of the problem is carried out to determine the design principles. When the pareto-optimal results are examined:  $c$ ,  $f$  and  $\delta$  values of Gripper I and  $b$ , and  $d$  values of Gripper II are changed; the other parameters has a fixed value. In addition, the problem specific performances of multiobjective optimizers applied to the problem are compared under equal conditions. MOAAA comes to the fore as the only method to obtain the best pareto-front curves for both Gripper I and Gripper II problems. These solutions are also the best-known results in the literature. The results of this study will be guiding for designers and researchers in use of metaheuristic algorithms for gripper design optimization. In future studies, it is planned to reformulate the robot gripper design problem by removing the remaining design variables from the configurations.

## References

- [1] Pham D, Heginbotham W. Robot Grippers. Bedford: IFS, 1986.
- [2] Tai K, El-Sayed AR, Shahriari M, Biglarbegian M, Mahmud S. State of the art robotic grippers and applications. *Robotics* 2016; 5(2): 11. doi:10.3390/robotics5020011
- [3] Chen FY. Gripping mechanisms for industrial robots: an overview. *Mechanism and Machine Theory* 1982; 17(5): 299-311. doi:10.1016/0094-114X(82)90011-8
- [4] Osyczka A, Krenich S, Karas K. Optimum design of robot grippers using genetic algorithms. In: *Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization (WCSMO)*; Buffalo, New York, USA; 1999. pp. 241-243.
- [5] Osyczka A. *Evolutionary Algorithms for Single and Multicriteria Design Optimization*. Berlin: Springer Verlag, 2002.
- [6] Saravanan R, Ramabalan S, Ebenezer NGR, Dharmaraja C. Evolutionary multi criteria design optimization of robot grippers. *Applied Soft Computing* 2009; 9(1): 159-172. doi:10.1016/J.ASOC.2008.04.001
- [7] Datta R, Deb K. Multi-objective design and analysis of robot gripper configurations using an evolutionary-classical approach. In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*; Dublin, Ireland; 2011. pp.1843-1850. doi:10.1145/2001576.2001823
- [8] Rao RV, Savsani VJ, Vakharia DP. Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *CAD Computer Aided Design* 2011; 43(3): 303-315. doi:10.1016/j.cad.2010.12.015
- [9] Datta R, Pradhan S, Bhattacharya B. Analysis and design optimization of a robotic gripper using multi-objective genetic algorithm. *Transactions on Systems, Man, and Cybernetics: Systems* 2016; 46(1): 16-26. doi:10.1109/TSMC.2015.2437847
- [10] Avder A, Şahin İ, Dörterler M. Multi-objective design optimization of the robot grippers with SPEA2. *International Journal of Intelligent Systems and Applications in Engineering* 2019; 7(2): 83-87. doi:10.18201/ijisae.2019252785
- [11] Cutkosky MR. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *Transactions on Robotics and Automation* 1989; 5(3): 269-279.
- [12] Bicchi A, Kumar V. Robotic grasping and contact: a review. In: *IEEE International Conference on Robotics and Automation*; San Francisco, CA, USA; 2000. pp.348-353. doi:10.1109/ROBOT.2000.844081
- [13] Lanni C, Ceccarelli M. An optimization problem algorithm for kinematic design of mechanisms for two-finger grippers. *The Open Mechanical Engineering Journal* 2009; 3(1): 49-62. doi:10.2174/1874155x00903010049

- [14] Li QM, Qin QH, Zhang SW, Deng H. Optimal design for heavy forging robot grippers. *Applied Mechanics and Materials* 2010; 44(47): 743-747. doi:10.4028/www.scientific.net/AMM.44-47.743
- [15] Dao TP, Huang SC. An optimal study of a gripper compliant mechanism based on Fuzzy-Taguchi method. *Applied Mechanics and Materials* 2013; 418: 141-144. doi:10.4028/www.scientific.net/AMM.418.141
- [16] Pahwa T, Gupta S, Bansal V, Prasad B, Kumar D. Analysis & design optimization of laterally driven poly-silicon electro-thermal micro-gripper for micro-objects manipulation. In: *COMSOL Conference*; Bangalore; 2012.
- [17] Abu Zitar RA. Optimum gripper using ant colony intelligence. *Industrial Robot: An International Journal* 2005; 32(1): 17-23. doi:10.1108/01439910510573246
- [18] Ciocarlie M, Hicks FM, Stanford S. Kinetic and dimensional optimization for a tendon-driven gripper. In: *2013 IEEE International Conference on Robotics and Automation*; Karlsruhe, Germany; 2013. pp.2751-2758. doi:10.1109/ICRA.2013.6630956
- [19] Holland, JH. *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, Massachusetts, USA: MIT press, 1992.
- [20] De Jong, K. Learning with genetic algorithms: an overview. *Machine Learning* 1988; 3(2-3): 121-138.
- [21] Whitley, D. A genetic algorithm tutorial. *Statistics and Computing* 1994; 4(2): 65-85.
- [22] Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of International Conference on Neural Networks*; Perth, WA, Australia; 1995. pp. 1942-1948. doi:10.1109/ICNN.1995.488968
- [23] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Advances in Engineering Software* 2014; 69: 46-61. doi:10.1016/J.ADVENGSOFT.2013.12.007
- [24] Uymaz SA, Tezel G, Yel E. Artificial algae algorithm (AAA) for nonlinear global optimization. *Applied Soft Computing Journal* 2015; 31: 153-171. doi:10.1016/j.asoc.2015.03.003
- [25] Babalik A, Ozkis A, Uymaz SA, Kiran MS. A multi-objective artificial algae algorithm. *Applied Soft Computing Journal* 2018; 68: 377-395. doi:10.1016/j.asoc.2018.04.009
- [26] Tawhid, MA, Savsani, V. A novel multi-objective optimization algorithm based on artificial algae for multi-objective engineering design problems. *Applied Intelligence* 2018; 48(10): 3762-3781. doi:10.1007/s10489-018-1170-x
- [27] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation* 1999; 3(4): 257-271. doi:10.1109/4235.797969
- [28] Bradstreet L. *The hypervolume indicator for multi-objective optimisation: calculation and use*. PhD, University of Western Australia, Perth, Australia, 2011.