Advances in Computational Design, *Vol. 4, No. 3 (2019) 197-210*
DOI: https://doi.org/10.12989/acd.2019.4.3.197

Synthesis of four-bar linkage motion generation using optimization algorithms

Wisanu Phukaokaew[1], Suwin Sleesongsom*[2], Natee Panagant[1] and Sujin Bureerat[1]

[1]Sustainable and Infrastructure Research and Development Center, Department of Mechanical Engineering, Faculty of Engineering, KhonKaen University, KhonKaen City, Thailand
[2]Department of Aeronautical Engineering, International Academy of Aviation Industry, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

(Received May 27, 2018, Revised January 3, 2019, Accepted January 13, 2019)

Abstract. Motion generation of a four-bar linkage is a type of mechanism synthesis that has a wide range of applications such as a pick-and-place operation in manufacturing. In this research, the use of meta-heuristics for motion generation of a four-bar linkage is demonstrated. Three problems of motion generation were posed as a constrained optimization probably using the weighted sum technique to handle two types of tracking errors. A simple penalty function technique was used to deal with design constraints while three meta-heuristics including differential evolution (DE), self-adaptive differential evolution (JADE) and teaching learning based optimization (TLBO) were employed to solve the problems. Comparative results and the effect of the constraint handling technique are illustrated and discussed.

Keywords: mechanism synthesis; four-bar linkage; motion generation; constraint handling; evolutionary algorithms

1. Introduction

A number of machines in daily life are the applications of a simple four-bar linkage (Acharyya and Mandal 2009, Samer *et al*. 2012, Tong *et al*. 2013, Yu *et al*. 2013, Sleesongsom and Bureerat 2015, Ebrahim and Payvandy 2015, Sleesongsom and Bureerat 2017, Sleesongsom and Bureerat 2018). Some applications for this mechanism is a crank-rocker, part of a train wheel, a window wiper, a door closing mechanism, and rock crushers etc. This is a reason why many researchers have studied on four-bar mechanism synthesis. The mechanism synthesis is categorized in two types as typical and dimensional syntheses. For the dimensional synthesis, it can be subdivided into three types, which are path generation, motion generation and function generation. In this research, we focus only on the path generation (Acharyya and Mandal 2009, Samer *et al*. 2012, Tong *et al*. 2013, Sleesongsom and Bureerat 2015, Ebrahim and Payvandy 2015, Sleesongsom and Bureerat 2017) and the motion generation (Samer *et al*. 2012, Peng 2010, Sleesongsom and Bureerat 2017). The path generation or path synthesis is a technique to find the dimensions of linkage that a point on a coupler link move along the desired path while the motion generation or motion synthesis has a set of desired points and desired angles as a goal (Acharyya and Mandal

*Corresponding author, Ph.D., E-mail: suwin.se@kmitl.ac.th

http://www.techno-press.org/?journal=acd&subpage=7 ISSN: 2383-8477 (Print), 2466-0523 (Online)

2009, Peng 2010). It has been proposed to study the path generation of the four bar mechanism by using meta-heuristic optimizers and studying the comparative performance.

Furthermore, the work by Ebrahim and Payvandy (2015) proposed the efficient constraint handling technique based on a penalty technique. Later the efficient technique to avoid the constraint related to a crank angle sequence was proposed by Sleesongsom and Bureerat (2015). This technique has been proved to enhance the performance of the path generation synthesis, thus, it is believed to be able to enhance the performance of the motion generation in this research. Very recent work by Sleesongsom and Bureerat (2018) focused on studying the new constraint handling technique, which is applied for the path generation problem. This technique is extended from their previous work. It is seen that the new technique outperforms the previous penalty technique.

It has been found that the advantages of using meta-heuristics (Baluja 1994, Storn and Price 1997, Acharyya and Mandal 2009, Zhangand Sanderson 2009, Rao and Patel 2012, Tong *et al*. 2013, Yu *et al*. 2013, Mirjalili *et al*. 2014, Sleesongsom and Bureerat 2015, Ebrahim and Payvandy 2015, Shaheen *et al*. 2015, Sleesongsom and Bureerat 2018) are robustness, simplicity to use and independence of function derivatives, however, they are infamous for the lack of a convergence speed and consistency. However, in the last decade, numerous meta-heuristics were developed, which mostly have been proven to outperform their predecessors in both convergence speed and consistency. As a result, we choose three of such meta-heuristics to study in this research including differential evolution (DE), self-adaptive differential evolution (JADE) and teaching learning based optimization (TLBO).

The rest of this paper is organized as follows. Section 2 details the position analysis of a four-bar linkage. Objective function, the constraint handling technique, and all of MHs are presented in Section 3. A design problem and its conditions as well as a numerical experiment are given in Section 4, while the design results are in Section 5. The conclusions and discussion of the study are summarized in Section 6.

2. Position analysis of a four-bar mechanism

A position analysis of a four-bar linkage is an important computation required for path synthesis. The kinematic diagram of a four-bar linkage is shown in Fig. 1.

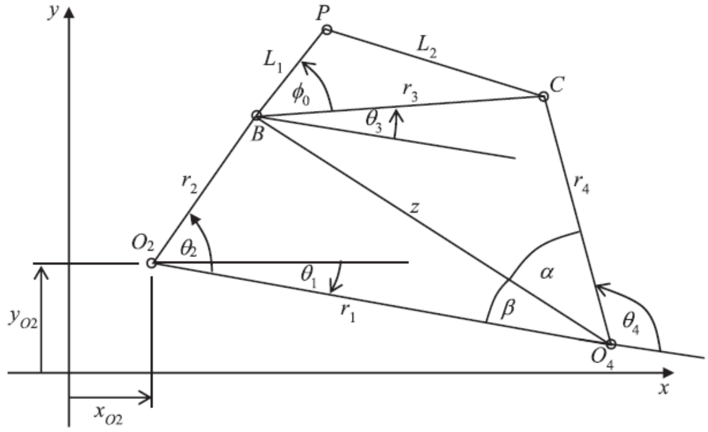


Fig. 1 Four-bar linkage in global coordinate system

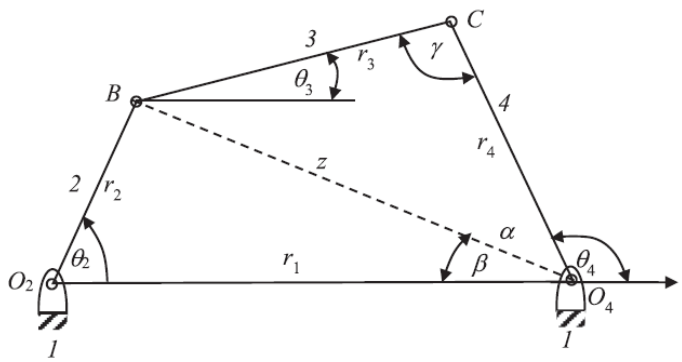


Fig. 2 Four-bar linkage in local coordinates

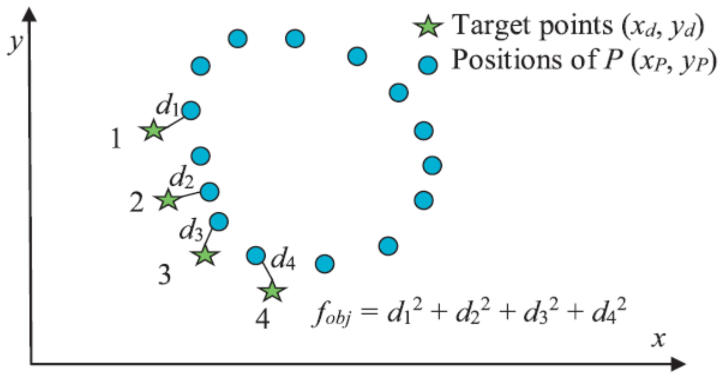


Fig. 3 Calculation of objective function value

For motion generation, it is assigned to search for link lengths (r_1, r_2, r_3, r_4) and other parameters, which gives the minimum error between the desire paths (x_d, y_d) and the generated points (x_p, y_p) and the minimum error between the desire angles (θ_{3d}) and generated angles (θ_{3p}) of the coupler link.

2.1 Position analysis

From Fig. 1, the vector loop equation of the mechanism can be written as

$$\boldsymbol{r}_1 + \boldsymbol{r}_2 + \boldsymbol{r}_3 + \boldsymbol{r}_4 = 0 \tag{1}$$

The coupler point coordinates in the global coordinate as shown in Fig. 1 can be expressed as

$$\begin{aligned} x_p &= x_{O2} + r_2\cos(\theta_2+\theta_1) + L_1\cos(\phi_0+\theta_3+\theta_1) \\ y_p &= y_{O2} + r_2\sin(\theta_2+\theta_1) + L_1\sin(\phi_0++\theta_3+\theta_1) \end{aligned} \tag{2}$$

where x_{O2} and y_{O2} are the coordinates of the O_2 pin joint in the global coordinates. And ϕ_0 can be obtained by considering the link *BCP* using the cosine law, which is expressed as

$$\phi_0 = \cos^{-1}\left[\frac{L_1^2 + r_3^2 - L_2^2}{2L_1 r_3}\right] \tag{3}$$

The values of angles θ_3, θ_4 and γ with the known link lengths r_1, r_2, r_3 and r_4 at any given crank angle (θ_2) can be obtained by considering Figure 2 and the technique in (Myszka 2005). The computation can be shown as follows:

$$Z^2 = r_1^2 + r_2^2 - 2r_1r_2\cos\theta_2$$

$$Z^2 = r_3^2 + r_4^2 - 2r_3r_4\cos\gamma$$

$$\gamma = \cos^{-1}\left[\frac{r_3^2 + r_4^2 - r_1^2 - r_2^2 + 2r_1r_2\cos\theta_2}{2r_3r_4}\right] \tag{4}$$

$$\gamma = \cos^{-1}\left[\frac{r_3^2 + r_4^2 - z^2}{2r_3r_4}\right]$$

$$\alpha = \cos^{-1}\left[\frac{z^2 - r_3^2 + r_4^2}{2zr_4}\right] \tag{5}$$

$$\beta = \cos^{-1}\left[\frac{z^2 + r_1^2 - r_2^2}{2zr_1}\right] \tag{6}$$

$$\theta_3 = \pi - (\alpha + \beta + \gamma) \tag{7}$$

$$\theta_4 = \pi - (\alpha + \beta) \tag{8}$$

These equations will be used for objective function evaluation of the proposed optimization problem.

3. Optimization problem and constraint handling

3.1 Optimization problem

The objective function composes two parts where the first part of the objective function is the position error between a set of desired points P_d or coordinates (x_d, y_d) initiated by the designer and the generated point (x_p, y_p) from the given mechanism dimensions. The second part of objective function is the angular error between desire angles (θ_{3d}) and generated angles (θ_{3p}) from input design variables. The member of design variables includes r_1, r_2, r_3, r_4, L_1, L_2, and the coordinates of O_2 (x_{O2}, y_{O2}) and the angle of frame *1* (θ_1). In this research, only the problem type called synthesis without prescribed timing, whicha set of $\theta_2{}^i$valuesare also set as design variables. The optimization problem without prescribed timing is then written as

$$\min f(\mathbf{x}) = \sum_{i=1}^{N}[(x_{d,i}-x_{p,i})^2 + (y_{d,i}-y_{p,i})^2 + (\theta_{3d,i}-\theta_{3p,i})^2] \tag{9}$$

subject to

$$\min(r_1, r_2, r_3, r_4) = \text{crank}(r_2) \tag{10}$$

$$2\min(r_1, r_2, r_3, r_4) + 2\max(r_1, r_2, r_3, r_4) < (r_1 + r_2 + r_3 + r_4) \tag{11}$$

$$\theta_2^1 < \theta_2^2 \ ... < \theta_2^N \tag{12}$$

$$\boldsymbol{x}_l \leq \boldsymbol{x} \leq \boldsymbol{x}_u. \tag{13}$$

where $\boldsymbol{x} = \{r_1, r_2, r_3, r_4, L_1, L_2, \theta_0, x_{O2}, y_{O2}, \theta_2^1, \theta_2^2, \dots, \theta_2^N\}^T$, N is the number of points on the prescribed or desired curve, and $\mathbf{x}_l$ and $\mathbf{x}_u$ are lower and upper bounds of design vector $\mathbf{x}$, respectively.

3.2 Constraint handling

This research uses a penalty function technique for constraint handling. This technique is carried out by adding to the objective function valued a penalty function value if the solution is infeasible. There are two parts of penalty function values. The first part is assigned to control link lengths to obey the Grashof's criterion as shown in equations (10) and (11). The second part of the penalty function is assigned to control the constraint (12) or to ensure the input crank can rotate with a complete revolution in either direction (clockwise or counter clockwise). The penalty function works with adding a very high value to modify the objective function when some of the constraints violate.

3.3 Meta-heuristics

The meta-heuristic algorithms used in this study are briefly detailed as:

3.3.1 Differential evolution (DE)

DE is one of the most popular and powerful MHs which is a population-based stochastic optimization method. It starts with an initial population, which is randomly generated when no preliminary knowledge about the solution space is available. The DE operators include mutation, crossover, and selection. These operators are used to maintain population diversity, as well as to avoid a premature convergence. The DE scheme used in this study can be classified as the standard DE/best/2/bin algorithm (Stornand Price 1997).

3.3.2 Self-adaptive differential evolution (JADE)

JADE is an optimizer with self-adaptive parameter settings of DE. It is regarded as one of the most powerful DEs. For example, the scaling factor (F) used in DE mutation and the crossover probability (CR) are crucial parameters for its performance and is also problem-dependent. As a consequence, the development of self-adaptive DE began, which to some extent can improve DE search performance. Adaptive schemes of JADE can update the control parameters based on their historical record of success (Zhang and Sanderson 2009).

3.3.3 Teaching learning based optimization (TLBO)

TLBO (Rao and Patel 2012) exploits the concept of teaching and learning behavior of a teacher and students in a classroom. Surely, all students will follow their teacher and they often learn from each other where the clever one will also teach another. With such an idea, TLBO is formulated in such a way that its reproduction process has two main operators namely teaching and learning phases. The algorithm is a population-based optimizer where a population of design solutions is improved iteratively until reaching the termination criterion.

Table 1 Response surface solutions

x1=F	0.5	0.5	0.5	1	1	1	1.5	1.5	1.5
x2=CR	0.7	0.8	0.9	0.7	0.8	0.9	0.7	0.8	0.9
Mean	0.7487	0.7514	0.7467	21.6982	20.6272	5.0437	61.6918	50.0431	80.5539

DE is chosen in this work because it is one of the most popular and most powerful meta-heuristics used in a wide variety of engineering applications while JADE is its self-adaptive version. TLBO, on the other hand, was the top performer for path synthesis (Sleesongsom and Bureerat 2017), thus, it is another preferred optimizer. Only DE requires optimization parameters setting. We thus performed DE by using design of experiment (DOE) to construct a response surface of the parameters settings (F and CR). The optimum parameters are used as DE control parameters for solving the design problems. The three-level full factorial design was performed leading to 3^2training points. For DOE, the scaling factor is set as F$\in$ {0.5, 1.0, 1.5}, while the crossover rate is assigned as CR $\in$ {0.7, 0.8, 0.9}. In this study, we use the weighted sum W_1= 0.25 and W_2 = 0.75 to tradeoff the two design objectives. The response surface solution is shown in Table 1, while the optimum parameters are F = 0.5 and CR = 0.8472. These parameters, given in Table 2, are employed with DE for performance comparison.

4. Numerical experiment

Nine design problems are used to study the performance of the meta-heuristics in this research. They are coded in MATLAB. The Optimizers parameters are tabulated in Table 2. The design problems are detailed as follows:

Case-1: Design variables are **x**
x= [r_1, r_2, r_3, r_4, L_1, L_2, x_{O2}, y_{O2},θ_1, $\theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5$, θ_2^6]
Target points are (x_d, y_d) and θ_{3d}
(x_d, y_d) = [(20,20), (20,25), (20,30), (20,35), (20,40), (20,45)]
θ_{3d}= [1.9937, 1.9220, 1.8434, 1.7599, 1.6709,1.5735] rad

Limits of the variables:
$5 \le r_1, r_2, r_3, r_4 \le 60$
$-60 \le L_1, L_2, x_{O2}, y_{O2}, \le 60$
$0 \le \theta_0, \theta_2^1, \dots, \theta_2^6 \le 2\pi$

Case-2: Design variables are **x**
x= [r_1, r_2, r_3, r_4, L_1, L_2, x_{O2}, y_{O2},θ_1, θ_2^1, $\theta_2^2,\theta_2^3,\theta_2^4,\theta_2^5$, θ_2^6, $\theta_2^7,\theta_2^8,\theta_2^9,\theta_2^{10}$]

Target points are (x_d, y_d) and θ_{3d}
(x_d, y_d) = [(20, 10), (17.66, 15.142), (11.736, 17.878), (5, 16.928), (0.60307, 12.736), (0.60307, 7.2638), (5, 3.0718), (11.736, 2.1215), (17.66, 4.8577), (20, 10)]
θ_{3d} = [0.4208, 0.5117, 0.7433, 0.9910, 1.1394, 1.1296, 0.9599, 0.7322, 0.5257, 0.4208] rad

Limits of the variables:
$5 \le r_1, r_2, r_3, r_4 \le 80$
$-80 \le L_1, L_2 x_{O2}, y_{O2}, \le 80$
$0 \le \theta_0, \theta_2^1, \ldots, \theta_2^{10} \le 2\pi$

Case-3 : Design variables are **x**
x= $[r_1, r_2, r_3, r_4, L_1, L_2, x_{O2}, y_{O2}, \theta_1, \theta_2^1, \theta_2^2, \theta_2^3, \theta_2^4, \theta_2^5, \theta_2^6, \theta_2^7, \theta_2^8, \theta_2^9, \theta_2^{10}, \theta_2^{11}, \theta_2^{12}, \theta_2^{13}, \theta_2^{14}, \theta_2^{15}, \theta_2^{16}, \theta_2^{17}, \theta_2^{18}, \theta_2^{19}, \theta_2^{20}, \theta_2^{21}]$
Target points are (x_d, y_d) and θ_{3d}
(x_d, y_d) = [(0, 0), (0.9356, 0.4064), (1.5983, 0.8220), (2.1704, 1.3002), (2.6621, 1.7986), (2.9957, 2.2692), (3.2614, 2.7276), (3.4941, 3.1408), (3.6635, 3.5485), (3.7798, 3.9623), (3.7992, 4.4734), (3.7170, 4.8783), (3.4675, 5.3624), (3.1242, 5.6980), (2.7158, 5.8800), (2.2517, 5.8780), (1.7997, 5.6882), (1.2732, 5.1741), (0.8336, 4.5628), (0.4003, 3.9064), (0.3, 3.38)]
θ_{3d} = [0, 0, 0, 0, 0, 0, 0.06984127, 0.13968254, 0.261904762, 0.436507937, 0.611111111, 0.838095238 1.047619048, 1.30952381, 1.571428571, 1.658730159, 1.746031746, 1.833333333, 1.746031746 1.658730159, 1.571428571] rad

Limits of the variables:
$5 \le r_1, r_2, r_3, r_4 \le 80$
$-80 \le L_1, L_2, x_{O2}, y_{O2}, \le 80$
$0 \le \theta_0, \theta_2^1, \ldots, \theta_2^{21} \le 2\pi$

The objective function equation (9) can be separated into two parts as the position errors and the angle error. Since there are two objective functions with different units, it is interesting to study their effects on motion synthesis. The objective function is then rewritten in the weighted sum form as:

$$f(\mathbf{x}) = W_1 \sum_{i=1}^{N} \left(x_{d,i} - x_{p,i}\right)^2 + \left(y_{d,i} - y_{p,i}\right)^2 + W_2 \sum_{i=1}^{N} \left(\theta_{3d,i} - \theta_{3p,i}\right)^2 \quad (14)$$

where W_1 and W_2 are weighting factors for position and angle errors respectively.
In this work, with the three original design problems and three sets of weighting factors, the optimization test problems are:

Case 1.
Case-11: W_1=0.5, W_2=0.5.
Case-12: W_1=0.25, W_2=0.75.
Case-13: W_1=0.75, W_2=0.25.

Case 2.
Case-21: W_1=0.5, W_2=0.5.
Case-22: W_1=0.25, W_2=0.75.
Case-23: W_1=0.75, W_2=0.75

Case 3.
Case-31: W_1=0.5, W_2=0.5.
Case-32: W_1=0.25, W_2=0.75.
Case-33: W_1=0.75, W_2=0.25.

Table 2 Parameters of the meta-heuristics

Parameters	DE	JADE	TLBO
Number of initial population	50	50	50
Scaling factor	0.5	-	-
Crossover rate	0.8472	-	-
Number of optimization run	30	30	30
Generation number	500	500	500

5. Results and discussion

Each meta-heuristic optimizer is used to solve the optimization problems for 30 runs where the best results are regarded as optimum solutions. The result of case-11 obtains from the three meta-heuristics are shown in Table 3. The table shows the best design result (x), mean objective function value (mean), best objective function value (min), standard deviation (std), number of successful runs (Success), and Error (exact objective function). It should be noted that the successful run means an optimization runs that results in a feasible optimum solution. In this case, there are 6 target points. It is found that TLBO gives the best error (error = 0.02), the second is DE (error = 0.0437) and the worst is JADE (error = 0.2036), nevertheless, DE gives the worst number of successful runs (80%) while the other algorithms have 100% (JADE) and 93.34% (TLBO) successful runs. When taking into account searching consistency based on the mean objective values and successful runs, the most consistent method is TLBO while the second best is DE.

For the case-12, the comparative results are reported in Table 3. It is found that DE (error = 0.0063) gives the better result than the other algorithms. The second best is TLBO (error = 0.0597) and the worst is JADE (error = 0.1345). However, JADE gives result in100.00% successful runs, while DE and TLBO gives the successful runs of 83.3333% and 96.67%, respectively. DE gives the worst number of successful runs. The most consistent method is JADE based on mean and successful runs.

In case-13, the results obtained by those algorithms are shown in Table 3. It is found that the DE algorithm gives the minimum error (error = 0.0137) while the second best is TLBO (error = 0.0536) and the worst in this case is JADE (error = 0.319). For this case the percentages of successful runs performed by DE, JADE and TLBO are 56.6667%, 96.67% and 83.34 %respectively. The most consistent method is TLBO based on mean and successful runs.

Figs. 4 show the best four-bar mechanisms obtained from solving Case-11, Case-12, and Case-13 respectively. The position errors for Case-11, Case-12, and Case-13 are 0.0150, 0.0060 and 0.0096 respectively. On the other hand, the angular errors for Case-11, Case-12, and Case-13 are 0.0053, 0.00029 and 0.0042respectively. It can be seen that they have different dimensions depending on the assigned weighting values.

The results of case-21 obtained from the three optimizers are shown in Table4. In this case (all of case-2) there are 10 target points. It is found that DE gives the best result (error = 0.7484) but with 33.3333 % successful runs. The second best is TLBO (error = 0.7833) with 36.67% successful runs while the worst is JADE (error = 1.905) with 10% successful runs. The most consistent method is DE based on mean and successful runs.

For Case-22, the results are shown in Table 4. It is found that DE (error = 0.7396) gives the best result (only minimum error) than the other algorithms. The second is TLBO (error = 0.7538) and

Table 3 Comparative results for Case-11-13

Parameter	Case-11			Case-12			Case-13		
	DE	JADE	TLBO	DE	JADE	TLBO	DE	JADE	TLBO
r_1	60	56.3425	14.1465	45.1775	26.1585	9.521	48.6865	31.532	9.961
r_2	42.4440	17.738	9.4605	31.4165	6.8755	5	32.0710	6.43	5.011
r_3	46.3655	32.9895	58.0585	37.0485	35.998	56.4305	41.0580	37.5875	57.569
r_4	56.0785	55.7815	60	44.4680	45.051	57.9705	49.7920	50.5895	59.2355
L_1	-17.6760	54.372	-50.76	44.0400	39.624	-39.408	59.9880	50.4	-41.94
L_2	57.9480	-59.952	-54.768	-35.8560	-45.24	-60	-36.7800	-44.568	-57.264
x_{O2}	-7.3920	56.964	-21.228	59.0880	-12.816	54.396	-24.9600	-23.652	56.904
y_{O2}	57.0960	16.692	49.056	9.5880	39.6	33.648	59.0640	47.76	31.692
θ_0	5.3325	6.2643	5.5707	0.5806	3.1661	2.7533	3.9113	3.3056	2.5818
θ_2^1	0.4105	0.1579	1.6103	3.5994	0.8841	5.0202	2.9679	1.1051	4.9255
θ_2^2	0.5052	0.2526	1.9891	3.4415	0.6315	4.4835	3.1258	0.7893	4.3888
θ_2^3	0.5999	0.3473	2.368	3.2837	0.4105	3.9783	3.2837	0.5368	3.8836
θ_2^4	0.6946	0.4420	2.7469	3.1258	0.1894	3.4415	3.4415	0.3157	3.3784
θ_2^5	0.7893	0.5368	3.1574	2.9679	6.2516	2.8732	3.5994	0.0631	2.8416
θ_2^6	0.8841	0.6315	3.5678	2.8101	5.9674	2.2733	3.7573	6.0937	2.3049
mean	0.1664	0.4174	0.2045	0.1605	0.1950	0.3639	0.18724	0.6185	0.3369
min	0.0214	0.1008	0.0092	0.0015	0.0799	0.0348	0.0080	0.1760	0.0172
max	1.1145	1.0028	1.0121	1.2990	0.4148	1.84219	1.5102	1.1902	1.8798
Std	0.2755	0.2207	0.2408	0.2945	0.0724	0.4475	0.3579	0.2577	0.4740
Success	24	30	28	25	30	29	17	29	25
Success(%)	80	100	93.34	83.3333	100	96.6667	56.6667	96.67	83.34
Exact Error	0.0437	0.2036	0.02	0.0063	0.1345	0.0597	0.0138	0.319	0.0536

*Success = no. of successful runs; error = the objective function (minimum error)

the worst optimizer is JADE (error = 1.8558). TLBO has 46.67% of number of successful runs, which is highest value. The second, DE has40% of number of successful runs while the rest have 23.34% successful runs. The most consistent method is TLBO according to the number of successful runs and the mean objective function values.

In case-23, the results obtained from using those algorithms are shown in Table 4. It is found that DE gives the minimum objective function (error = 0.7505) while the second best and the worst are TLBO (error = 0.7546) and JADE (error = 2.7638) respectively. The table shows the number of successful runs for DE as 36.6667 % and for TLBO and JADE are 20% and 6.67% respectively. DE is the best method for both convergence and consistency in this case.

Figs. 5 shows the best four-bar mechanisms obtained from solving Case-21, Case-22, and Case-23 respectively. The position errors for Case-21, Case-22, and Case-23 are 0.0275, 0.0169, and 0.0292 respectively. On the other hand, the angular errors for Case-21, Case-22, and Case-23 are 0.7210, 0.7227, and 0.7213respectively. It can be seen that they have different dimensions depending on the assigned weighting values.

The comparative results of case-31 are reported in Table 5 where the number of target points is

Table 4 Comparative results for Case-21-23

Parameter	Case-21			Case-22			Case-23		
	DE	JADE	TLBO	DE	JADE	TLBO	DE	JADE	TLBO
r_1	73.1825	36.0125	79.955	54.2900	63.1625	73.6475	80	55.79	77.63
r_2	10.0100	9.755	9.0125	9.2375	9.44	9.08	8.8400	10.1375	9.3875
r_3	43.2800	42.2225	64.5275	35.6150	55.835	47.1125	55.4675	46.37	52.175
r_4	44.5100	29.4725	52.7825	32.1950	42.59	45.4625	49.8875	43.295	49.07
L_1	7.9200	-6.304	-14.24	7.1200	12.544	9.616	11.9680	9.776	10.576
L_2	-35.9040	-38.704	-66.112	36.5760	-46.4	49.488	59.8400	-37.056	-50.384
x_{O2}	17.8880	15.584	0.128	4.6240	20.816	16.816	17.4400	19.632	1.328
y_{O2}	9.4400	8.064	20.128	5.6960	16.064	16.592	19.2320	10.16	4.144
θ_0	2.1627	2.9933	0.0666	4.7840	2.3166	1.5658	1.5350	2.0351	4.9204
θ_2^1	4.1046	3.2521	6.0937	1.5787	4.073	4.8308	4.8624	4.3572	1.4524
θ_2^2	4.8308	3.8836	0.5368	2.2733	4.7992	5.5254	5.5570	5.0202	2.147
θ_2^3	5.5254	4.6098	1.2314	2.9995	5.4938	6.22	6.2516	5.6517	2.8416
θ_2^4	6.2200	5.3044	1.926	3.7257	6.1569	0.6315	0.6630	0.0631	3.5678
θ_2^5	0.6315	6.0622	2.6206	4.4203	0.5368	1.3261	1.3577	0.8841	4.2625
θ_2^6	1.3261	0.5052	3.3152	5.1150	1.2314	2.0207	2.0523	1.5471	4.9571
θ_2^7	1.9891	1.1682	4.0099	5.8096	1.9576	0.4276	2.7469	0.6357	5.6517
θ_2^8	2.6838	1.8629	4.7045	0.1894	2.6522	0.4976	3.4731	0.6738	0.0631
θ_2^9	3.3784	2.6206	5.3991	0.8841	3.3784	0.6349	4.1677	0.8026	0.7578
θ_2^{10}	4.1046	3.2521	6.0937	1.5787	4.073	0.7699	4.8624	0.9581	1.4524
mean	0.6596	1.2640	0.8792	0.7455	1.2892	0.9790	0.6615	1.7659	0.8290
min	0.3711	0.9395	0.3900	0.5461	0.8190	0.5490	0.2005	1.5339	0.2032
max	1.7411	1.6399	1.8387	1.3966	1.7272	1.5955	1.9825	1.9979	1.7529
Std	0.50355	0.3531	0.6118	0.3481	0.3365	0.3780	0.6005	0.3281	0.6262
Success	10	3	11	12	7	14	11	2	6
Success(%)	33.3333	10	36.67	40	23.34	46.67	36.6667	6.67	20
Exact Error	0.7485	1.905	0.7833	0.7396	1.8558	0.7538	0.7506	2.7638	0.7546

21. It is found that DE gives the best result (error = 12.9063), the second is TLBO (error = 13.3751) and the worst is JADE (error = 17.5293). For this case the percentages of successful runs performed by DE, JADE and TLBO are 26.6667%, 20% and 60 % respectively. TLBO is the best method for both convergence and consistency in this case.

For case-32, the results obtained from the three optimizers are shown in Table 5. It is found that the best solution is from using DE (error = 13.5807), the second is TLBO (error =14.6835) and the worst is JADE (error = 19.3363). DE, JADE and TLBO are 40%, 16.67% and 60%of successful runs respectively. TLBO is the best method for both convergence and consistency in this case.

For Case-33, the results obtained from using the various MHs are shown in Table 5. From the results, it is found that DE gives the best result (error = 13.3806) but it is the second worst when considering the number of successful runs (23.3333%). The second best is TLBO (error = 14.299) which gives the best result for the number of successful runs while the worst error (error

Table 5 Comparative results for Case-31-33

Parameter	Case-31			Case-32			Case-33		
	DE	JADE	TLBO	DE	JADE	TLBO	DE	JADE	TLBO
r_1	79.9250	32.3675	22.82	51.2675	79.3325	52.775	79.9025	51.5825	41.8625
r_2	5	6.2075	5.5475	5	6.305	5	7.6775	9.665	5
r_3	26.1575	53.8775	44.4875	80	47.045	79.4675	30.5300	26.7725	50.645
r_4	58.7750	48.74	35.21	58.4750	49.37	58.76	57.0500	36.41	35.12
L_1	13.1680	-28.48	-25.696	-65.5200	36	-68.144	18.8320	21.472	-37.904
L_2	-28.3680	45.536	-37.584	-78.2240	24.816	-80	29.2000	17.44	-51.936
x_{O2}	4.1440	-11.28	-3.936	53.3920	-4.784	-37.936	5.1840	7.488	-28.544
y_{O2}	-7.3760	-18.864	-24.56	46.0640	37.872	-54.432	-10.1760	-12.896	-16.448
θ_0	6.0400	0.9494	1.4809	4.1306	3.7435	1.2598	0.0038	0.4769	1.0725
θ_2^1	4.9887	3.0942	4.6098	5.8412	6.2832	4.1046	4.7992	3.7573	2.7153
θ_2^2	5.3991	3.5994	4.6729	5.3991	6.22	4.1993	5.1465	3.6626	3.2837
θ_2^3	5.6201	3.9152	1.2945	5.1781	6.1569	1.1998	5.3675	3.5994	3.5363
θ_2^4	5.8096	4.073	1.1682	4.9887	2.4628	1.0735	5.4938	5.5886	3.7257
θ_2^5	5.9674	4.2309	1.0419	4.8624	2.5259	0.9472	5.6201	5.6517	3.852
θ_2^6	6.0622	4.3572	0.9472	4.7361	2.5891	0.8525	5.7149	5.6833	3.9783
θ_2^7	6.1569	4.4519	0.8525	4.6413	2.6838	0.7578	5.7780	5.7464	4.073
θ_2^8	6.2516	4.5151	0.7578	4.5466	2.7153	0.663	5.8412	5.778	4.1677
θ_2^9	0.0631	4.6098	0.663	4.4519	2.7785	0.5683	5.9043	5.8412	4.2309
θ_2^{10}	0.1263	4.6729	0.5683	4.3888	2.8416	0.4736	5.9674	5.8727	4.294
θ_2^{11}	0.2210	4.7676	0.442	4.2940	2.9364	0.3789	6.0306	5.9043	4.3888
θ_2^{12}	0.2842	4.7992	0.3473	4.2309	2.9995	0.2842	6.0622	5.9359	4.4519
θ_2^{13}	0.3473	4.8624	0.1579	4.1677	3.0942	0.1263	6.1253	5.999	4.5151
θ_2^{14}	3.2205	4.9255	6.2516	1.3892	3.1574	0	3.3784	6.0306	4.5466
θ_2^{15}	3.2521	1.5787	6.0937	1.3577	3.2205	6.0937	3.4100	6.0306	1.0104
θ_2^{16}	3.3152	1.6418	5.9359	1.2945	3.2205	5.557	3.4415	6.0306	1.0419
θ_2^{17}	3.3784	1.705	5.8096	1.2314	5.3675	5.3675	3.5047	2.9679	1.1367
θ_2^{18}	3.5047	1.7997	5.6201	1.0735	5.4623	5.1781	3.5994	3.0311	1.263
θ_2^{19}	3.6310	1.926	5.4623	0.9472	5.5886	4.9887	3.6941	3.0942	1.3892
θ_2^{20}	3.7573	2.0839	5.3044	0.7893	5.6833	4.8308	3.8204	3.1574	1.5471
θ_2^{21}	3.8520	2.1786	5.2097	0.6946	5.778	4.7361	3.8836	3.2205	1.6418
mean	7.5227	10.9663	9.0400	9.5873	11.4091	10.5664	6.0078	10.5556	7.5320
min	6.44931	8.7754	6.6875	9.2129	10.7913	9.4850	3.9664	8.7715	4.7757
max	8.8288	11.8902	11.8350	10.1972	11.9699	11.5793	7.7882	11.6537	10.6825
Std	0.7161	1.1535	1.3830	0.2772	0.4359	0.5658	1.5665	1.1430	2.0439
Success	8	6	18	12	5	18	7	5	12
Success(%)	26.6667	20	60	40	16.67	60	23.3333	16.67	40
Exact Error	12.9064	17.5293	13.3751	13.5802	19.3363	14.6835	13.3805	20.6548	14.299

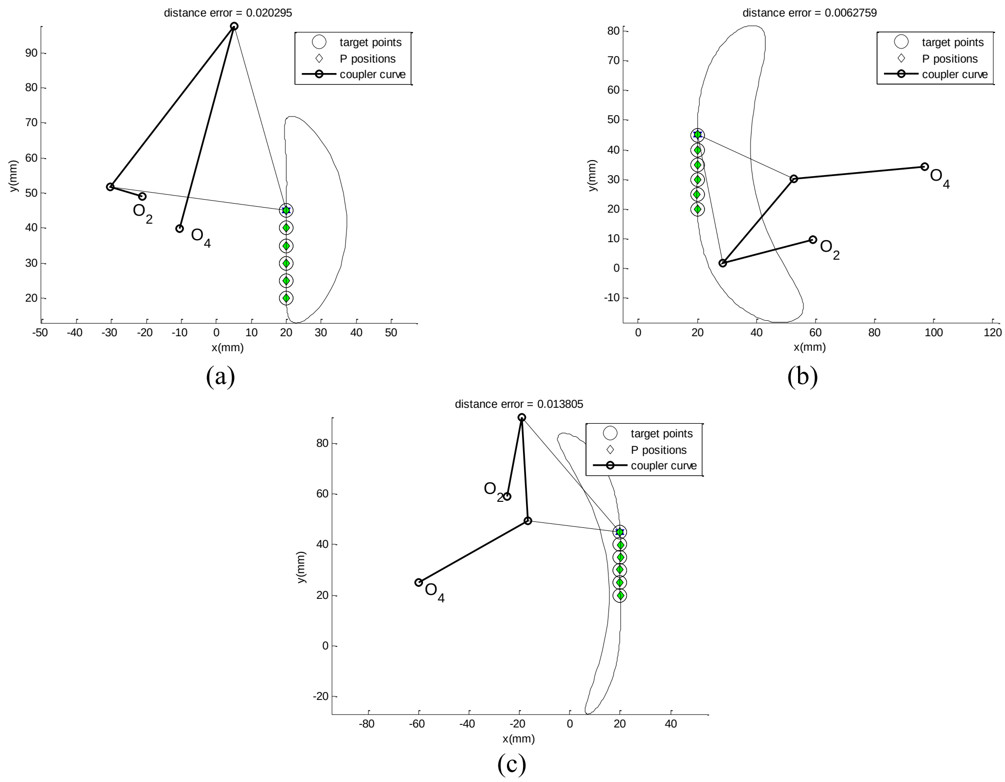


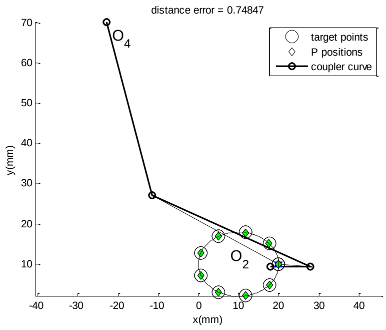
Fig. 4 Best solution (a) Case-11 (TLBO) (b) Case-12 (DE) (c) Case-13 (DE)

=20.6548) and worst successful runs (16.67%) are from JADE respectively. TLBO is the best method for both convergence and consistency in this case.

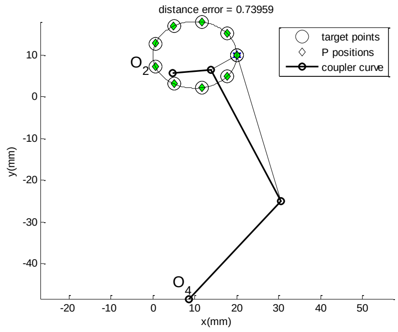
Figs. 6 show the best four-bar mechanisms obtained from solving Case-31, Case-32, and Case-33 respectively. The position errors for Case-31, Case-32, and Case-33 are 0.9746, 1.9434, and 1.2329respectively. On the other hand, the angular errors for Case-31, Case-32, and Case-33 are 11.9319, 11.6368, and 12.1477 respectively. It can be seen that they have different dimensions depending on the assigned weighting values.

6. Conclusions

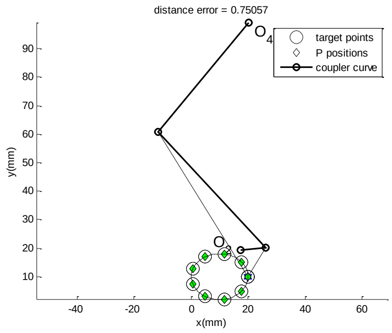
The comparative results reveal that the employed meta-heuristics can be used to solve the motion synthesis problems for a four-bar linkage successfully. Overall, DE/best/2/bin gives the best solutions for most of the design cases with TLBO being the second best. Nevertheless, when considering the search consistency, TLBO is an algorithm responding best to this criterion. JADE is not efficient for this type of optimization. The weighting factors influent the results, which implies that it is up to a designer to pre-specify whether to weight to position or angular errors. This study reports the baseline results for other researchers to follow and develop more powerful algorithms to solve the test problems. It is also more interesting to use multiobjective evolutionary algorithms to deal with minimizing position and angular errors. With the use of such optimisers, a designer does not need to predefine weighting factors and a Pareto front can be achieved within one optimization run.



(a)

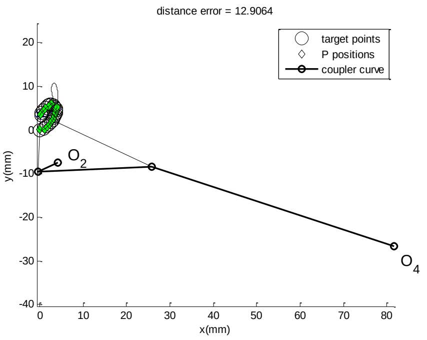


(b)

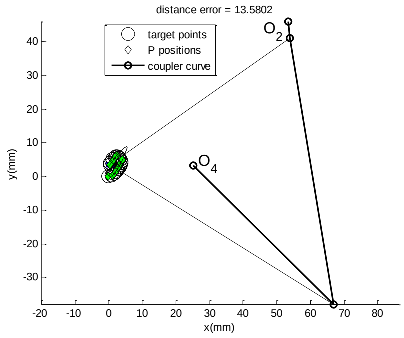


(c)

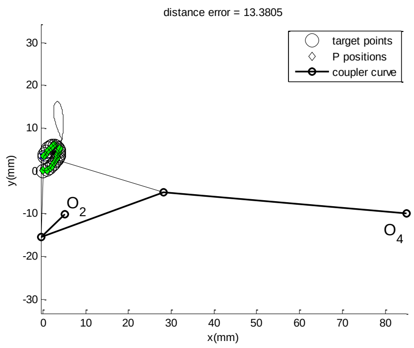
Fig. 5 Best solution for (a) Case-21 (DE) (b) Case-22 (DE) (c) Case-23 (DE)



(a)



(b)



(c)

Fig. 6 Best solution for (a) Case-31 (DE) (b) Case-32 (DE) (c) Case-33 (DE)

Acknowledgments

The authors are grateful for the financial support provided by the Thailand Research Fund.

References

Acharyya, S.K. and Mandal, M. (2009), "Performance of EAs for four-bar linkage synthesis", *Mech. Mach. Theory*, **85**, 1784-1794. https://doi.org/10.1016/j.mechmachtheory.2009.03.003.

Baluja, S. (1994), "Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning", CMU-CS-95-163; Carnegie Mellon University, USA.

Ebrahimi, S. and Payvandy, P. (2015), "Efficient constrained synthesis of path generating four-bar mechanisms based on the heuristic optimization algorithms", *Mech. Mach. Theory*, **85**, 189-204. https://doi.org/10.1016/j.mechmachtheory.2014.11.021.

Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014). "Grey wolf optimizer", *Adv. Eng. Softw*, **69**, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

Mutawe, S.K., Al-Smadi, Y.M. and Sodhi, R.S. (2012), "Designing four-bar linkages for path generation with worst case joint clearances", *Eng. Lett.*, **20**(2), 143-147.

Myszka, D.H. (2005), *Machines and Mechanisms: Applied Kinematic Analysis, (3rd Edition)*, Prentice Hall, New Jersey, USA.

Peng, C. (2010), "Optimal synthesis of planar adjustable mechanism", Ph.D. Dissertation, New Jersey Institute of Technology, New Jersey, USA.

Rao, R.V. and Patel, V. (2012), "An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems", *Sci. Iran. D*, **20**(3), 710-720. https://doi.org/10.1016/j.scient.2012.12.005.

Shaheen, A.M., Spea, S.R., Farrag, S.M. and Abido, M.A. (2015), "A review of meta-heuristic algorithms for reactive power planning problem", *Ain Shams Eng. J.*, **9**(2), 215-231. https://doi.org/10.1016/j.asej.2015.12.003.

Sleesongsom, S. and Bureerat S. (2018), "Alternative constraint handling technique for four-bar linkage path generation", *IOP Conf. Series: Mater. Sci. Eng.*, **324**, https://doi.org/10.1088/1757-899X/324/1/012012.

Sleesongsom, S. and Bureerat, S. (2015), "Morphing wing structural optimization using opposite-based population-based incremental learning and multigrid ground elements", *Math. Probl. Eng.*, **2015**, http://dx.doi.org/10.1155/2015/730626.

Sleesongsom, S. and Bureerat, S. (2015), "Optimal synthesis of four-bar linkage path generation through evolutionary computer", *Res. Appl. Mech. Eng.*, **3**(2), 46-53. https://doi.org/10.1155/2018/5462563.

Sleesongsom, S. and Bureerat, S. (2017), "Four-bar linkage path generation through self-adaptive population size teaching-learning based optimization", *Knowl.-Based Syst.*, **135**, 180-191. https://doi.org/10.1016/j.knosys.2017.08.012.

Storn, R.M. and Price, K.V. (1997), "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces", *J. Global Optim*, **11**, 341-359. https://doi.org/10.1023/A:1008202821328.

Tong, Y., Myszka, D.H. and Murray, A.P. (2013), "Four-bar linkage synthesis for a combination of motion and path-point generation", *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Oregon, USA, August.

Yu, H.Y., Zhao, Y.W., and Wang, Z.X. (2013), "Study on numerical comparison method of four-bar straight-line guidance mechanism", *J. Harbin Inst. Technol.*, **20**, 63-72.

Zhang, J. and Sanderson, A.C. (2009), "JADE: adaptive differential evolution with optional external archive", *IEEE Transaction on Evol. Comp.*, **13**(5), 945-958. https://doi.org/10.1109/TEVC.2009.2014613.