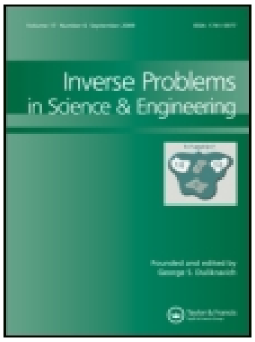


Taylor & Francis
Taylor & Francis Group

Inverse Problems in Engineering

ISSN: 1068-2767 (Print) 1029-0281 (Online) Journal homepage: http://www.tandfonline.com/loi/gipe19

# Optimum Robot Design Based on Task Specifications Using Evolutionary Techniques and Kinematic, Dynamic, and Structural Constraints

P.S. Shiakolas , D. Koladiya & J. Kebrle

**To cite this article:** P.S. Shiakolas , D. Koladiya & J. Kebrle (2002) Optimum Robot Design Based on Task Specifications Using Evolutionary Techniques and Kinematic, Dynamic, and Structural Constraints, Inverse Problems in Engineering, 10:4, 359-375, DOI: 10.1080/1068276021000004706

**To link to this article:** http://dx.doi.org/10.1080/1068276021000004706

Published online: 18 Oct 2010.

*Inverse Problems in Engng*, 2002, Vol. 10, No. 4, pp. 359–375

# OPTIMUM ROBOT DESIGN BASED ON TASK SPECIFICATIONS USING EVOLUTIONARY TECHNIQUES AND KINEMATIC, DYNAMIC, AND STRUCTURAL CONSTRAINTS

P.S. SHIAKOLAS*, D. KOLADIYA and J. KEBRLE

*Mechanical and Aerospace Engineering Department, The University of Texas at Arlington, Arlington, TX 76019-0023, USA*

*(Received 26 March 2002; In final form 15 May 2002)*

In this article, we discuss optimum robot design based on task specifications using evolutionary optimization approaches. The three evolutionary optimization approaches employed are Simple Genetic Algorithms, Genetic Algorithms with elitism, and Differential Evolution (DE). These approaches were used for the optimum design of SCARA and articulated type manipulators based on kinematic, dynamic and structural analyses. The objective function minimizes the torque required for the motion subject to deflection and physical constraints with the design variables being the link physical characteristics (length and cross-sectional area parameters). In this work, we experimented with various cross-sections for the links. The main findings of this research are that the DE converges quickly, requires significantly less number of iterations and achieves better results by reaching smaller objective functions.

*Keywords:* Robot design; Optimization; Evolutionary techniques; Dynamic; Kinematic; Structural

## 1. INTRODUCTION

In this article, we discuss optimum robot design based on task specifications using three evolutionary optimization approaches. The three evolutionary optimization approaches employed are simple genetic algorithm (SGA), genetic algorithm with elitism (GAE), and differential evolution (DE). These approaches were used for the optimum design of SCARA and articulated type manipulators. The objective function minimizes the torque subject to deflection and physical constraints for a defined manipulator motion with the design variables being the link physical characteristics (length and cross-sectional area parameters). The links of serial manipulators are usually over designed in order to be able to support the subsequent links on

*Corresponding author. Tel.: (817) 272-5715. Fax: (817) 272 2952. E-mail: shiakolas@mae.uta.edu

ISSN 1068-2767 print: ISSN 1029-0281 online © 2002 Taylor & Francis Ltd
DOI: 10.1080/1068276021000004706

the chain and the payload to be manipulated. However, increasing the size of the links unnecessarily requires the use of larger actuators resulting in higher power requirements.

Optimum robot design has been addressed by many researchers as found in the open literature based on kinematic and/or dynamic specifications. Oral and Idler optimized a planar robotic arm for a particular motion. They described the minimum weight design of high-speed robot using sequential quadratic programming [1]. Yang and Ming-Chen applied the minimized degree of freedom concept in a modular robotic system in order to maximize the load carrying capacity and reduce the power requirements using an evolutionary algorithm (a variation of GA) [2]. Paredis and Khosla discussed optimum kinematic design for serial link manipulators using simulated annealing based on task specifications [3]. Tang and Wang compared Lagrangian and primal–dual neural network for real time joint torque optimization of kinematically redundant manipulators. They used desired acceleration of end-effector as input for a specific task [4]. Chen and Cheng described various control schemes and explained advantages of "minimum velocity norm method" over other methods for local joint torque optimization in redundant manipulator [5]. Paredis used a distributed agent based genetic algorithm (GA) to create fault tolerant serial chain manipulators from a small inventory of links and modules [6]. Chedmail and Ramstein used a GA to determine the base position and type (one of several) of a manipulator to optimize workspace reachability [7].

The major disadvantage of conventional optimization techniques is that they work from point to point, using local information to decide which point to explore next. This could lead to the premature convergence of optimization process at false optimum. Also, many of the conventional search techniques require specific knowledge of the problem to be analyzed, for example gradient-based techniques required derivative information. Another drawback of traditional optimization techniques is the requirement for good initial guesses for the design variables.

Evolutionary based techniques are employed in order to overcome shortcomings of conventional optimization methods [2,8–10]. GA and their variants are part of the evolutionary techniques and have been extensively used for modular robot design [11–14], inverse and forward kinematics [9,15] and optimum motion and path planning [10,16]. DE is a recently developed evolutionary technique [17,18]. DE has been successfully applied to the optimum design of digital filter and communication control [17], and according to information on the web [19] to many diverse domains such as dynamic systems, controller design, heat transfer analysis and design, and many more. The DE technique to the best of our knowledge has never been investigated or applied to the optimum design of serial link manipulators.

This article addresses the application and comparison of three evolutionary techniques for optimum design of serial link robotic manipulators based to task specifications. The objective function minimizes the required torque for a defined motion subject to various constraints while considering kinematic, dynamic and structural conditions. The kinematic and dynamic analyses are derived based on robotic concepts and the structural analysis is performed based on the finite element method. The design variables examined are the link parameters and the link cross-sectional characteristics. The developed environment was employed in optimizing the design variables for a SCARA and an articulated 3-DOF PUMA type manipulators.

## 2. OPTIMIZATION PROBLEM DEFINITION

The general optimization problem is defined as follows [20]

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } g_i(x) \leq 0 \end{aligned} \tag{1}$$

where $f(x)$ is the objective function, $g_i(x)$ is the set of inequality constraints and $x \in \Re^n$is the real-valued design variable vector with $n$ being the number of design variables.

The optimization problem investigated is that of optimizing the link parameters for a robotic manipulator to obtain minimum power requirements or joint torques. The task specification is subdivided into two elements, the kinematic characteristics (the required position of the end effector) and dynamic characteristics (the time required to complete the motion while carrying the payload and considering the inertia properties of the links themselves). Constraints are imposed on the minimum and maximum (range) values of the link parameters that include the link length, and the link cross-sectional area characteristics, and the allowable deflection of the end effector. In addition, kinematics constraints are imposed that address physical limitations such as the range of motion of the actuators of the manipulator.

- Objective Function

The objective function is defined as the cumulative sum of the torques for each joint during the motion of the manipulator,

$$f(x) = \sum_{i=1}^{\text{time}} \sum_{j=1}^{\text{joints}} t_{ij}^2 \tag{2}$$

where $t_{ij}$ is the torque at time $i$ for joint $j$.

- Constraints

The constraints for this optimization include the deflection of the end effector of the manipulator, physical constraints such as the limits on the joint values, and the structural characteristics of each link. Other constraints that could be included for the design of manipulators are quantitative measures of various workspace attributes such as manipulability or the structural length index.

The constraints for this optimization include the deflection of the end effector of the manipulator, physical constraints such as the limits on the joint values, and the structural characteristics of each link.

$$\delta \leq \delta_{\text{allowable}} \tag{3}$$

The deflection $\delta$ is evaluated using finite element analysis techniques when the manipulator is at its maximum reach (completely stretched out) since this will yield the maximum deflection. Any other configuration will yield a smaller deflection value considering that the same payload is carried. The deflection evaluation is a function of the structural and material properties of the links and the payload.

The constraints on the joint limits or range of motion of the manipulator actuators are imposed due to physical constraints. The joint constraints are defined as

$$\theta_{i,\min} \leq \theta_i \leq \theta_{i,\max} \tag{4}$$

where $\theta_i$ is the joint value for joint $i$ and $\theta \in \Re^n$ with $n$ being the number of joints. This constraint is important in selecting a unique solution in the cases where the inverse kinematics solution process (a function of the Cartesian position and link length) yields multiple solutions.

The structural characteristics of the links are also included in the constraints. These structural characteristics are the link lengths and the cross-sectional area characteristics. These constraints are defined as

$$d_{j,\min} \leq d_j \leq d_{j,\max} \tag{5}$$

where $d_j$ indicates the $j$ structural characteristic, and $d \in \Re^m$ with $m$ being the number of structural characteristics. The design space for the structural characteristics always consists of the link length with the remaining parameters depended on the type of cross-section being considered. In this work, the cross-sections considered are hollow rectangular, square, cylindrical and C-channel and are shown in Fig. 1. The number of the structural constraints is dynamic and depends on the type of cross-section being considered in the analysis. In addition, physical structural constraints are imposed during the analysis in the sense that the inner cross-sectional dimensions cannot be equal or larger than the outer dimensions shown in Fig. 1. For example,

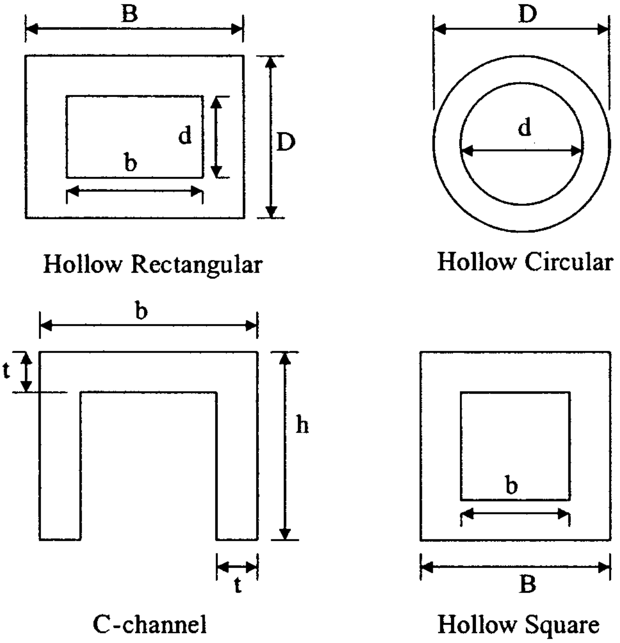


FIGURE 1 Cross-sections considered in the analysis.

if a manipulator is to be designed with a link having a C-channel cross-section, then the additional dynamic constraints will include $t < h$ and $t < b/2$.

- Analysis Procedure

The first step in the analysis process is to define the problem, define the design variables, assign values to all the parameters and define the constraint vector. The problem definition consists of defining the kinematic structure of the manipulator to be analyzed using Denavit–Hartenberg (DH) parameters, the desired initial and final position in Cartesian space and desired time for the motion, the payload, and the material property and cross-section type of the links. The schematic of this process is shown in Fig. 2.

The defined values are used in the analysis routines to obtain values for the design variables. The design variables are checked for constraint violation and then used in evaluating the objective function. These evaluations are used in the optimization routine where new values for the design variables are generated. One function evaluation is completed when one set of design variables is analyzed. The calculated joint torque provides a fitness evaluation for optimization. The fitness evaluation process is presented later. This process is repeated until certain criteria are met such as the number of generations.

The analysis routines related to manipulator motion analysis are inverse kinematics, joint space trajectory generation, and dynamics. These are evaluated using standard robotic analysis algorithms [21].

- Inverse kinematics evaluates the required joint variables $\theta_{\text{initial}}$ and $\theta_{\text{final}}$ based on the desired Cartesian coordinates of the initial and final positions.

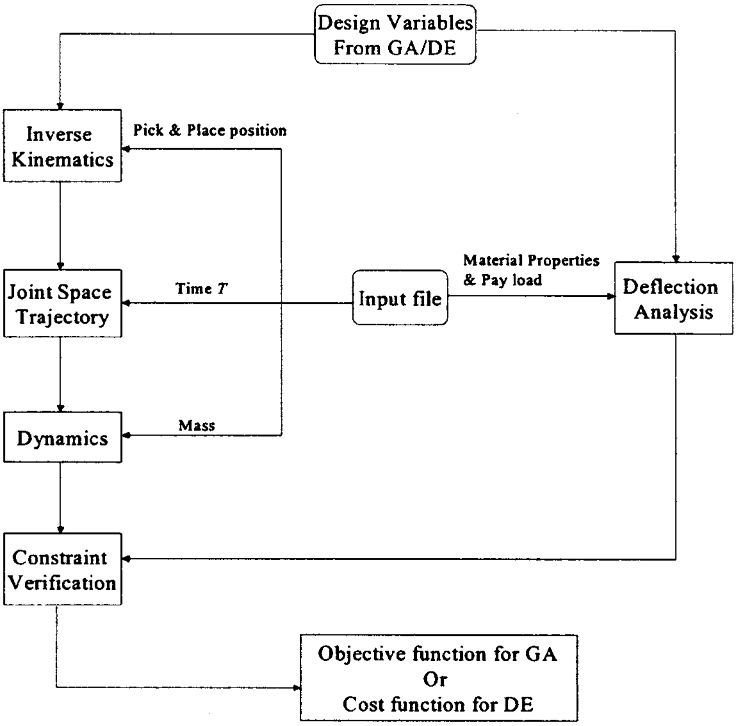


FIGURE 2 Analysis flow chart.

- Joint space trajectory generates the position $\theta(t)$ (cubic polynomial profile), velocity $\theta'(t)$ and acceleration $\theta''(t)$ profiles based on the desired motion time and joint variables.
- Dynamics evaluates the individual joint torques based on the structural characteristics of the links, the payload and the position, velocity and acceleration profiles. The evaluation of the dynamic equations could include also friction losses proportional to the magnitude of the velocity. In this work, the friction losses are not considered in the analysis since the friction coefficients are not known and their influence will be to increase the value of the torque required for the motion. The dynamics are evaluated according to the Newton–Euler method [21].

$$\tau(t) = M(\theta)\theta'' + V(\theta, \theta')\theta' + G(\theta) \tag{6}$$

- Finite Element Analysis for Deflection

The deflection analysis evaluates the deflection of the end effector of the manipulator considering the structural characteristics of the links, the material properties for inertia evaluation and the payload. This calculation is performed using the finite element method [22]. The maximum deflection occurs when the manipulator is at its maximum reach. The generalized load vector is derived using the payload, $F_p$, and the properties of the links of the manipulator such as cross-section, modulus of elasticity, $E$, and density, $\rho$.

The links of the manipulator are modeled as beams with the general 3-D beam element stiffness matrix [22]. The structural model of a two link planar manipulator is shown in Fig. 3 where the degrees of freedom for each node are the moments, $M_i$ due to bending about the $z$-axis and forces $f_i$ due to axial loading along the $y$-axis.

The assembled global load vector, $P$, for the two-element manipulator after reduction due to motion constraints at degrees of freedom $M_0$ and $f_0$ is given by

$$P = \begin{bmatrix} \frac{1}{2}(f_1 L_1 + f_2 L_2) \\ \frac{1}{12}(L_2^2 f_2 - L_1^2 f_1) \\ \frac{L_2 f_2}{2} + F_p \\ \frac{-f_2 L_2^2}{12} \end{bmatrix} \tag{7}$$

where $f_i$ is the link weight per unit length given by $f_i = (\rho_i A_i L_i g)/L_i$.

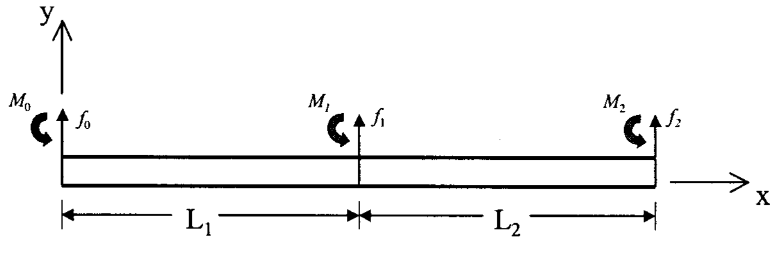


FIGURE 3 Forces and moments for structural analysis.

The stiffness matrices for each element are assembled to generate the global stiffness matrix. The assembled global stiffness matrix for the two-element manipulator, $K$, after reduction due to constraint degrees of freedom is given by

$$K = \begin{bmatrix} 12\left(\frac{E_1I_1}{L_1^3}+\frac{E_2I_2}{L_2^3}\right) & 6\left(\frac{E_2I_2}{L_2^2}-\frac{E_1I_1}{L_1^2}\right) & -12\left(\frac{E_2I_2}{L_2^3}\right) & 6\left(\frac{E_2I_2}{L_2^2}\right) \\ 6\left(\frac{E_2I_2}{L_2^2}-\frac{E_1I_1}{L_1^2}\right) & 4\left(\frac{E_1I_1}{L_1}+\frac{E_2I_2}{L_2}\right) & -6\left(\frac{E_2I_2}{L_2^2}\right) & 2\left(\frac{E_2I_2}{L_2}\right) \\ -12\left(\frac{E_2I_2}{L_2^3}\right) & -6\left(\frac{E_2I_2}{L_2^2}\right) & 12\left(\frac{E_2I_2}{L_2^3}\right) & -6\left(\frac{E_2I_2}{L_2^2}\right) \\ 6\left(\frac{E_2I_2}{L_2^2}\right) & 2\left(\frac{E_2I_2}{L_2}\right) & -6\left(\frac{E_2I_2}{L_2^2}\right) & 4\left(\frac{E_2I_2}{L_2}\right) \end{bmatrix} \tag{8}$$

The deflection vector $v = \{y_1, \theta_1, y_2, \theta_2\}$ is then evaluated according to

$$U = K^{-1}P \tag{9}$$

## 3. OPTIMIZATION APPROACHES

Three evolutionary based optimization approaches were investigated, a SGA, simple GAE and DE.

### 3.1. Genetic Algorithms

GAs are a stochastic (probabilistic) optimization approach whose search method is based on the theory of "survival of the fittest". It uses term individuals (or genotype, chromosomes) in a population. GAs spur the parallel search between many peaks and exploit the search space. This characteristic makes GA more robust as compared to other point-to-point search methods.

The GA input data are population size, number of generations, number of bits per variable, crossover probability and mutation probability. The population $P(t)$ can be expressed in a vector form,

$$\begin{aligned} P(t) &= [X_1^t, X_2^t, X_3^t, X_4^t, \ldots, X_P^t] \\ X_i^t &= [x_1, x_2, x_3, \ldots, x_d] \end{aligned} \tag{10}$$

where $t$ is the generation number, $X_i^t$ is the set of design variables, and $d$ is the number of design variables.

During each generation, chromosome $X_i$ is evaluated for fitness according to the ability of attaining the satisfactory result. The fitness function $F$ is calculated according to

$$\begin{aligned} F &= (Z - f) + (1 - I)Z \\ I &= \begin{cases} 1 & \text{if any constraint is violated} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{11}$$

where $f$ is the objective function, and $Z$ is a value greater than the objective function. The selection of this fitness function heavily penalizes chromosomes that violate any of the constraints. A chromosome that violates any of the constraints becomes very small as compared to the chromosome that satisfies all constraints. This is because the term $(1-I)Z$ in the objective function evaluation goes to zero in case of any constraint violation. GAs are characterized by the genetic operations of selection, crossover and mutation [23].

*Selection:* Using roulette wheel fit individuals are selected for genetic operations. Individuals having beneficial fitness are at higher chances for selection.

*Crossover:* For each chromosome in population, if the randomly generated number $r <$ Pc (probability of crossover), then that chromosome is selected for crossover. Two such selected vectors exchange their string information in hope of better children. Crossover occurs at a randomly generated number within number of bits and this point is called point of crossover.

*Mutation:* For each chromosome in the population, if the randomly generated number $r <$ Pm (probability of mutation), then the chromosome is selected for mutation. Mutation is a process that changes the binary structure of a randomly selected bit: from 0 to 1 or vice versa. This offers diversity to the search direction and prevents convergence to local minima.

These genetic operations generate a new population with higher average fitness. The newly created population replaces the old one and the iterative procedure repeats till the specific convergence criteria are met, usually, the number of generations. The control parameters for the SGA and GAE approaches are listed in Table I. The values for these parameters were selected after experimentation using the developed tools.

The flow of operations for optimization using GAs is shown in Fig. 4.

The SGA verifies newly generated children during the crossover operation in order to prevent a bad solution from entering into the new population. If the fitness of a child is lower than that of the parent, then the child is not permitted into the new population. The SGA could loose good members during the genetic operation of selection.

The GAE is similar to the SGA with an additional feature called "elitist strategy". The elitism prevents loosing the best results from the current generation due to the genetic operations of selection, crossover and mutation [24]. The objective of GAE is to freeze or forward the best solution of the previous generation to the next generation. The GAE fitness function $F$ is calculated according to Eq. (11).

### 3.2. Differential Evolution

The DE approach contains the same processes of population initialization, mutation, crossover and selection. It emphasizes direct use of the objective function. A cost

TABLE I GA control parameters

| | *SGA* | *GAE* |
|---|---|---|
| Crossover probability | 0.3 | 0.6 |
| Mutation probability | 0.06 | 0.04 |
| Population size | 60 | 45 |
| No. of generations | 50 | 30 |

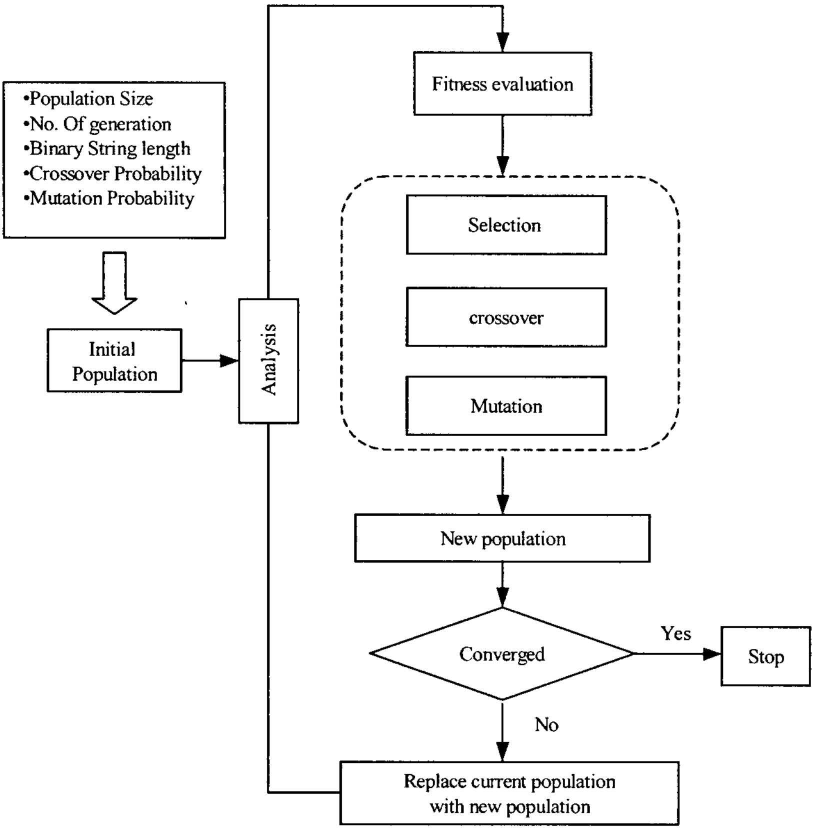


FIGURE 4 General flow of operations using GAs for optimization.

function, $C$, is used to rate the individual vectors according to their capability to minimize the objective function, $f$.

$$C = f + I$$
$$I = \begin{cases} 0 & \text{if no constraint violation} \\ > f & \text{otherwise} \end{cases} \tag{12}$$

Solution vectors without constraint violations have cost functions equal to the objective function. Any constraint violation tends to increase the cost of a vector by a value $I$ greater than the objective function. For our study of torque minimization, the lower the cost, the better or more fit the design variables are.

The genetic operation of mutation in the DE uses the vector differentiation method (adding the weighted difference between two population vectors to the third vector) to generate a new vector. DE is a parallel search method that operates on $D$-dimensional parameter vectors, where $D$ is the number of design variables. The number of vectors is equal to user defined population size. The initial vector population is chosen randomly. The DE process starts from selecting a target vector. Then, it randomly selects two other vectors and generates a difference vector, which is multiplied with a user defined weighting factor $F$ to obtain "weighted difference vector". The weighted difference vector and randomly chosen mutation vector create a noisy vector, which is subjected to crossover process with the target vector in order to generate the trial vector [17,25]. The trial vector is then compared with the original target vector. The vector having less cost function is allowed into the new population. A schematic of the DE process is shown in Fig. 5.

One of the advantages of DE is that it uses real variables as compared to binary numbers like the GA. This scheme encodes variables as floating point numbers without ever being reformatted as chromosomes with a different binary representation. Floating-point numbers use the computer resources efficiently [18]. In our work, we used the control parameters suggested by Storn [26] and shown in Table II. The population size is generally varied according to the number of design variables.

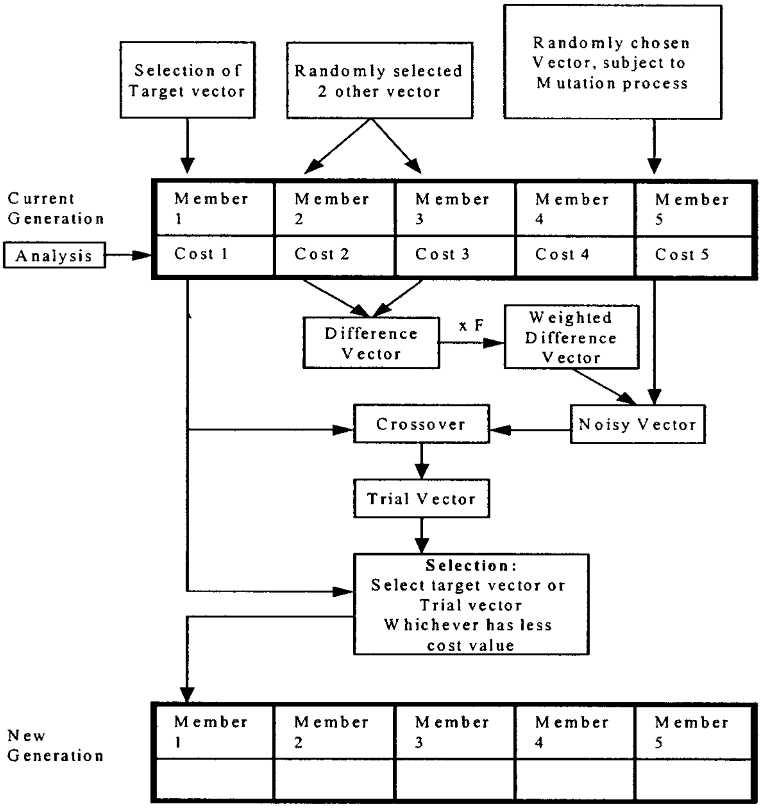

Selection of Target vector
Randomly selected 2 other vector
Randomly chosen Vector, subject to Mutation process
Current Generation
Analysis
Member 1
Member 2
Member 3
Member 4
Member 5
Cost 1
Cost 2
Cost 3
Cost 4
Cost 5
Difference Vector
x F
Weighted Difference Vector
Crossover
Noisy Vector
Trial Vector
Selection: Select target vector or Trial vector Whichever has less cost value
New Generation
Member 1
Member 2
Member 3
Member 4
Member 5

FIGURE 5 Schematic of Differential Evolution process [25].

TABLE II Differential Evolution control parameters [26]

| | |
|---|---|
| Crossover probability | 0.5 |
| Weighing factor | 0.8 |
| Population size | 60 |
| No. of generations | 20 |

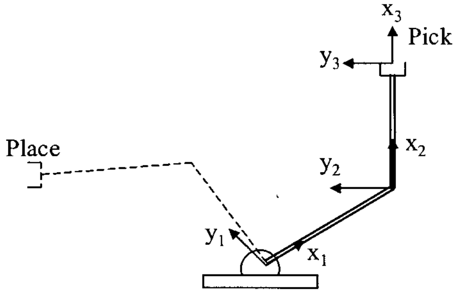
## 4. APPLICATION TO MANIPULATOR DESIGN

The evolutionary techniques introduced were used in the design of SCARA and articulated manipulators. In this section, the results of applying these techniques for manipulator design will be presented and discussed. The manipulator models, kinematics, inverse kinematics and dynamic analyses for both configurations are based on the modified DH parameters [21]. In the presented examples, the parameters are expressed in the SI system and the angles in degrees.

### 4.1. SCARA Manipulator

A SCARA planar manipulator with two revolute joints along with the modeling DH parameters are shown in Fig. 6. The task specification and constraints are given in Table III. The constraint values for the design variables, the link lengths and cross-sectional parameters for each cross-section, were defined to be the same for all three optimization approaches.

The results of the optimization process utilizing the three evolutionary techniques are presented in Tables IV–VII. In these tables, the values of the design variables for each cross-section as well as the torque and number of function evaluations for each technique are presented. The length values obtained satisfy the kinematic requirements (the manipulator reaches the desired points without violating any constraints) and along with the link parameters (cross-section and Young's modulus) the structural requirements. It is observed that the GAE and DE results are closer to each other as compared to those from the SGA approach. However, the DE approach consistently obtains smaller torques and smaller number of function evaluations. The number of function evaluations is an indication of the computing effort required in reaching the optimum torque value for the same number of generations.



DH-TABLE

| Frame | α | a | d | θ |
|---|---|---|---|---|
| 0-1 | 0 | 0 | 0 | $\theta_1$ |
| 1-2 | 0 | $L_1$ | 0 | $\theta_2$ |
| 2-3 | 0 | $L_2$ | 0 | 0 |

FIGURE 6 SCARA manipulator schematic and kinematic parameters.

TABLE III SCARA robot task specifications and constraints

| *Task* | | | | *Constraints* | |
|---|---|---|---|---|---|
| *Initial position* (m) | *Final position* (m) | *Motion time* (s) | *Payload* (kg) | *Max. deflection* (m) | *Joint range* (*Degrees*) |
| $X=0.3$<br>$Y=0.4$ | $X=-0.2$<br>$Y=0.3$ | 2.0 | 2.0 | 0.00001 | $0 \le \theta_1 \le 220$<br>$0 \le \theta_2 \le 160$ |

TABLE IV Circular cross-section (SI units)

| | *L1* | *L2* | *D1* | *D2* | *d1* | *d2* | *Torque *10E3* | *Function evaluations* |
|---|---|---|---|---|---|---|---|---|
| SGA | 0.2073 | 0.3025 | 0.0883 | 0.0583 | 0.0281 | 0.0281 | 7.90 | 9777 |
| GAE | 0.2143 | 0.2947 | 0.0863 | 0.0552 | 0.0227 | 0.0360 | 5.38 | 4801 |
| DE | 0.2292 | 0.2708 | 0.0736 | 0.0557 | 0.0233 | 0.0364 | 3.10 | 1260 |

TABLE V Square cross-section (SI units)

| | *L1* | *L2* | *D1* | *D2* | *d1* | *d2* | *Torque *10E3* | *Function evaluations* |
|---|---|---|---|---|---|---|---|---|
| SGA | 0.1881 | 0.3161 | 0.0808 | 0.0436 | 0.0159 | 0.0161 | 5.94 | 9777 |
| GAE | 0.2877 | 0.2216 | 0.0677 | 0.0374 | 0.0146 | 0.0192 | 5.35 | 4862 |
| DE | 0.2221 | 0.2798 | 0.0666 | 0.0420 | 0.0141 | 0.0186 | 3.75 | 1200 |

TABLE VI C-channel cross-section (SI units)

| | *L1* | *L2* | *H1* | *H2* | *B1* | *B2* | *T1* | *T2* | *Torque *10E3* | *Function evaluations* |
|---|---|---|---|---|---|---|---|---|---|---|
| SGA | 0.3360 | 0.1822 | 0.0780 | 0.0610 | 0.0699 | 0.0655 | 0.0131 | 0.0131 | 5.50 | 9826 |
| GAE | 0.2512 | 0.2612 | 0.0827 | 0.0591 | 0.0858 | 0.0659 | 0.0101 | 0.0102 | 3.34 | 5008 |
| DE | 0.2955 | 0.2046 | 0.0697 | 0.0512 | 0.0710 | 0.0748 | 0.0134 | 0.0107 | 2.77 | 1600 |

TABLE VII Rectangular cross-section (SI units)

| | *L1* | *L2* | *B1* | *B2* | *D1* | *D2* | *b1* | *b2* | *d1* | *d2* | *Torque *10E3* | *Function evaluations* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SGA | 0.2388 | 0.2706 | 0.0718 | 0.0485 | 0.0760 | 0.0459 | 0.0342 | 0.0350 | 0.0330 | 0.0342 | 3.71 | 9936 |
| GAE | 0.2541 | 0.2509 | 0.0527 | 0.0423 | 0.0787 | 0.0500 | 0.0309 | 0.0338 | 0.0337 | 0.0363 | 2.19 | 4835 |
| DE | 0.2174 | 0.2827 | 0.0491 | 0.0434 | 0.0862 | 0.0421 | 0.0350 | 0.0329 | 0.0345 | 0.0280 | 1.68 | 2000 |

TABLE VIII SCARA summary results (SI units)

| | *Circular cross-section* | | *Square cross-section* | | *C-channel cross-section* | | *Rectangular cross-section* | |
|---|---|---|---|---|---|---|---|---|
| | *Torque *10E3* | *Function evaluations* | *Torque *10E3* | *Function evaluations* | *Torque *10E3* | *Function evaluations* | *Torque *10E3* | *Function evaluations* |
| SGA | 7.90 | 9777 | 5.94 | 9777 | 5.50 | 9826 | 3.71 | 9936 |
| GAE | 5.38 | 4801 | 5.35 | 4862 | 3.34 | 5008 | 2.19 | 4835 |
| DE | 3.10 | 1260 | 3.75 | 1200 | 2.77 | 1600 | 1.68 | 2000 |

The optimization results summarizing the torque and function evaluations for the SCARA manipulator for the various cross-sections are shown in Table VIII. The rectangular cross-section provides minimum torque for the defined task without violating any of the constraints. The objective function history for the SCARA manipulator for each cross-section is shown in Fig. 7. It is observed that the DE converges faster than the other two in about 20 generations and in a monotonic fashion.

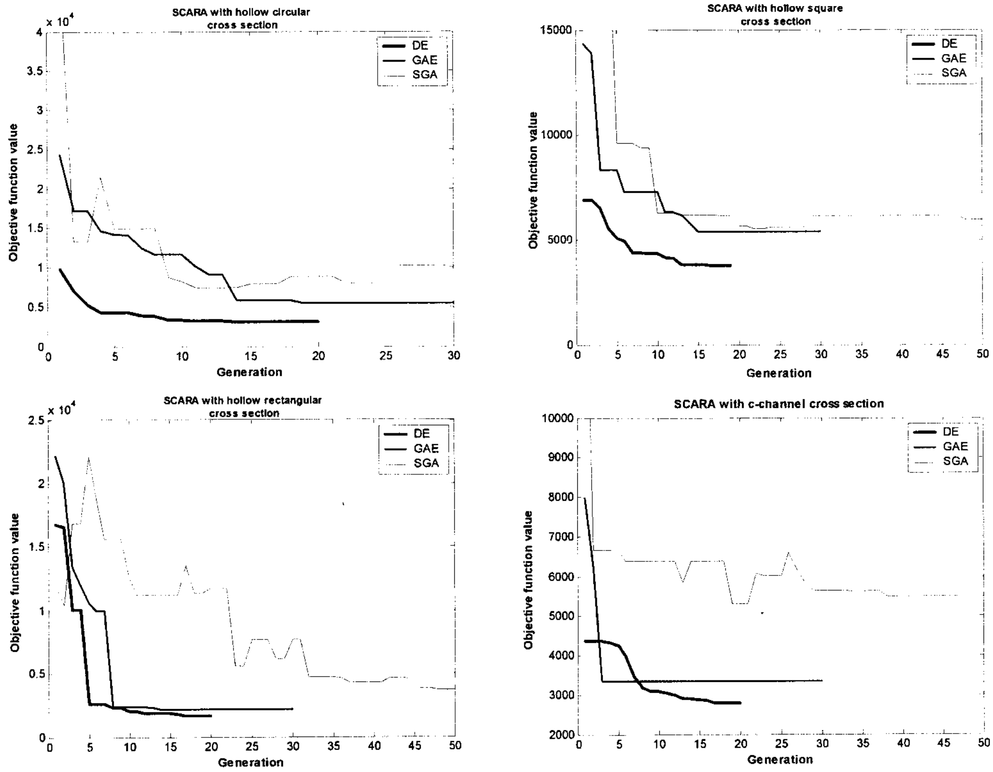


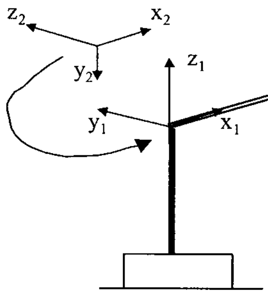
FIGURE 7 SCARA manipulator objective function history.

## 4.2. 3-DOF Articulated

A 3-DOF articulated PUMA type manipulator along with the modeling DH parameters are shown in Fig. 8. The last two links were optimized but the objective function included the torque from all three joints. The task specification and constraints are given in Table IX. The constraint values for the design variables were the same for all three optimization approaches.

The results of the optimization process utilizing the three evolutionary techniques are presented in Tables X–XII. In these tables, the values of the design variables for each cross-section and the torque and number of function evaluations for each technique are presented. The length values obtained satisfy the kinematic requirements (the manipulator reaches the desired points without violating any constraints) and along with the link parameters (cross-section and Young's modulus) the structural requirements. It is observed that the DE approach consistently obtains smaller torques and smaller number of function evaluations. The number of function evaluations is an indication of the computing effort required in reaching the optimum torque.

The optimization results summarizing the torque and function evaluations for the articulated manipulator for the various cross-sections are shown in Table XIII. The square cross-section provides for the minimum torque for the defined task while



DH-TABLE

| Frame | α | a | d | θ |
|---|---|---|---|---|
| 0-1 | 0 | 0 | 0 | $\theta_1$ |
| 1-2 | -90 | 0 | 0 | $\theta_2$ |
| 2-3 | 0 | $L_1$ | 0 | $\theta_3$+90 |
| 3-4 | 0 | $L_2$ | 0 | 0 |

FIGURE 8 Articulated PUMA type manipulator schematic and DH parameters.

TABLE IX Articulated manipulator task specifications and constraints

| *Task* | | | | *Constraints* | |
|---|---|---|---|---|---|
| *Initial position* (m) | *Final position* (m) | *Motion time* (s) | *Payload* (kg) | *Max. deflection* (m) | *Joint range* (*Degrees*) |
| $X=0.2$<br>$Y=0.2$<br>$Z=0.25$ | $X=-0.2$<br>$Y=0.25$<br>$Z=0.3$ | 2 | 2 | 0.00001 | $-135\leq\theta_1\leq135$<br>$0\leq\theta_2\leq-90$<br>$0\leq\theta_3\leq-90$ |

TABLE X Articulated circular cross-section (SI units)

| | *L1* | *L2* | *D1* | *D2* | *d1* | *d2* | *Torque* | *Function evaluations* |
|---|---|---|---|---|---|---|---|---|
| SGA | 0.3417 | 0.1461 | 0.1297 | 0.0520 | 0.0105 | 0.0398 | 428 | 9819 |
| GAE | 0.3271 | 0.1117 | 0.1033 | 0.0531 | 0.0113 | 0.0396 | 263.4 | 4811 |
| DE | 0.3345 | 0.1080 | 0.0773 | 0.0502 | 0.0336 | 0.0393 | 168 | 1200 |

TABLE XI Articulated square cross-section (SI Units)

| | *L1* | *L2* | *D1* | *D2* | *d1* | *d2* | *Torque* | *Function evaluations* |
|---|---|---|---|---|---|---|---|---|
| SGA | 0.1498 | 0.2957 | 0.0843 | 0.0399 | 0.0148 | 0.0170 | 622 | 9847 |
| GAE | 0.3084 | 0.1395 | 0.0786 | 0.0336 | 0.0070 | 0.0194 | 234 | 4762 |
| DE | 0.3256 | 0.1130 | 0.0700 | 0.0319 | 0.0134 | 0.0171 | 146 | 1200 |

TABLE XII Articulated rectangular cross-section (SI units)

| | *L1* | *L2* | *B1* | *B2* | *D1* | *D2* | *b1* | *b2* | *d1* | *d2* | *Torque* | *Function evaluations* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SGA | 0.3447 | 0.1296 | 0.0906 | 0.0478 | 0.0696 | 0.0452 | 0.0330 | 0.0269 | 0.0248 | 0.0347 | 671 | 9887 |
| GAE | 0.3468 | 0.1039 | 0.0510 | 0.0460 | 0.1015 | 0.0460 | 0.0214 | 0.0338 | 0.0214 | 0.0225 | 526 | 4865 |
| DE | 0.3289 | 0.1158 | 0.0500 | 0.0436 | 0.0972 | 0.0435 | 0.0226 | 0.0319 | 0.0319 | 0.0339 | 130 | 2000 |

TABLE XIII Articulated manipulator summary results in SI units

| | *Circular cross-section* | | *Square cross-section* | | *Rectangular cross-section* | |
|---|---|---|---|---|---|---|
| | *Torque* | *Function evaluations* | *Torque* | *Function evaluations* | *Torque* | *Function evaluations* |
| SGA | 428 | 9819 | 622 | 9847 | 671 | 9887 |
| GAE | 263 | 4811 | 234 | 4762 | 526 | 4865 |
| DE | 168 | 1200 | 146 | 1200 | 130 | 2000 |

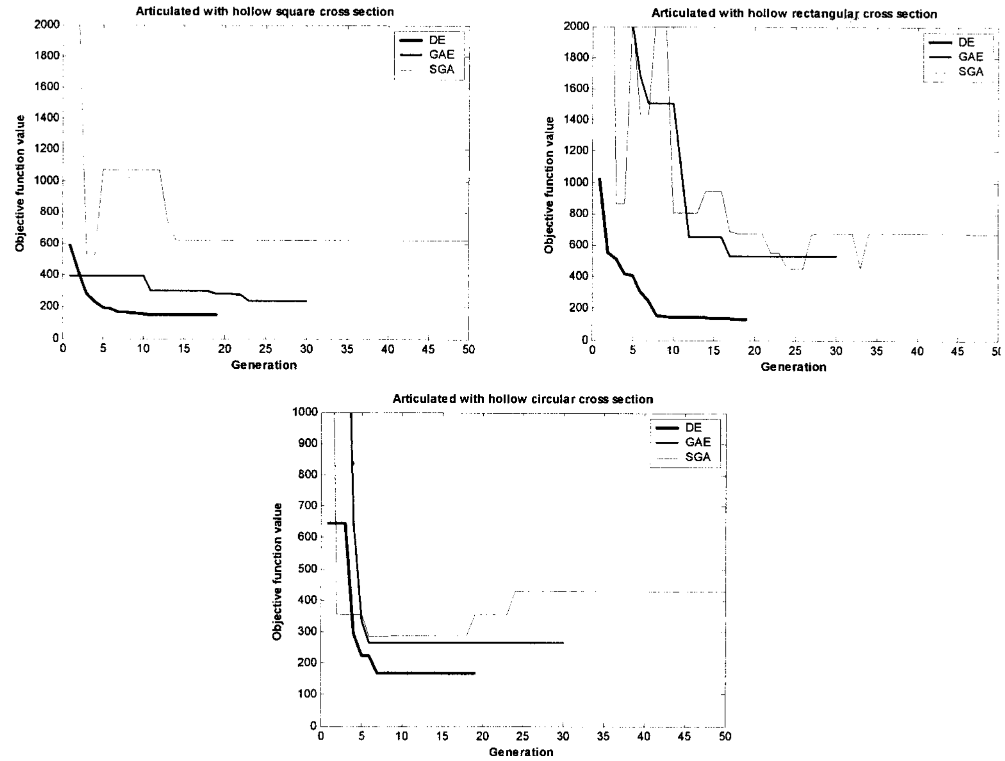


FIGURE 9 Articulated manipulator objective function history.

meeting the constraints when compared for all three approaches. The objective function history for the articulated manipulator for each cross-section is shown in Fig. 9. It is observed that the DE technique converges faster than the other two in a monotonic fashion and requires about 20 generations.

### 4.3. Additional Discussion of Results

The results presented show the objective function value (torque value) and number of function evaluations for each evolutionary approach for each cross-section analyzed.

It is observed that the torque value, the objective function used during the optimization process, for each cross-section is consistent between the three approaches. The smallest is obtained from DE followed by GAE (GA with elitism) with the largest value given by the simple GA. This indicates that the traditional GA techniques do not minimize the objective function or design variables even with a larger number of generations.

Another interesting observation is that the number of function evaluations follows a similar trend with the DE reaching a steady state or convergence type of behavior in only 20 generations. Even though the traditional GA techniques were allowed to run for more generations thus increasing the number of function evaluations, their objective functions were not even close to those obtained by DE.

The large difference in the number of function evaluations required to minimize the objective function is directly related to time savings. These time savings are amplified considering that the DE operates on real-valued design variables and does not require binary conversion as the GA.

## 5. CONCLUSIONS

In this article, three evolutionary optimization approaches, SGA, GAE and DE, were used and compared for the design of serial link robotic manipulators (SCARA and articulated) based on task specifications and constraints. The design process considered the kinematic, dynamic and structural characteristics of the links of the manipulator and the end-effector payload. The objective was to minimize the torque required to perform the defined motion subject to constraints on link parameters (length, and cross-sectional area), and end-effector deflection. These evolutionary techniques were applied in the design of SCARA and articulated type robotic devices using circular, square, rectangular and C-channel cross-sections for the links. The results indicate that the DE reaches a "steady-state" objective function quickly requiring only a small number of generations and function evaluations as compared to the other two. The DE optimization even with smaller number of generations evaluated design variables that yield significantly smaller objective functions. The SGA and GAE could not reach the same objective functions even when the number of generations was substantially increased. The DE technique operates on real-valued design variables and does not require conversions between binary and decimal. The fast performance of DE indicates that it could be a viable optimization approach for real-time optimization.

### *Acknowledgments*

The authors would like to thank Dr. Storn for making the DE code used in this study freely available [19] and his input on DE related questions. The authors would also like to thank Mr. Trogos of Mathworks Inc. for providing the MATLAB software used in this study.

### *References*

[1] S. Oral and S.K. Ider (1997). Optimum design of high-speed flexible robotic arms with dynamic behavior constraints. *Computer and Structures*, **65**(2), 255–259.

[2] G. Yang and I-Ming Chen (2000). Task based optimization of modular robot configurations: minimized degree-of-freedom approach. *Mechanism and Machine Theory*, **35**, 517–540.

[3] J.J. Paredis and P.K. Khosla (1993). Kinematic design of serial link manipulators from task specifications. *International Journal of Robotics Research*, **12**(3), 274–287.

[4] W.S. Tang and J. Wang (2000). Two recurrent neural networks for local joint torque optimization of kinematically redundant manipulators. *IEEE Transaction on Systems, Man and Cybernetics – Part B: Cybernetics*, **30**(1), 120–128.

[5] F.T. Cheng, T.H Chen and Y.Y Sun (1994). Torque optimization schemes for kinematically redundant manipulators. *Journal of Robotic Systems*, **11**(4), 257–269.

[6] J.J. Paredis and P.K. Khosla (1997). Agent-based design of fault tolerant manipulators for satellite docking. *Proceedings of IEEE International Conference on Robotics and Automation*, **4**, 3473–3480.

[7] P. Chedmail and E. Ramstein (1996). Robot mechanism synthesis and genetic algorithms. *Proceedings of IEEE International Conference on Robotics and Automation*, **4**, 3466–3471.

[8] J. Han, W.K. Chung, Y. Youm and S.H. Kim (1997). Task based design of modular robot manipulator using efficient genetic algorithm. *Proceedings of IEEE International Conference on Robotics and Automation*, **1**, 507–512.

[9] J.K. Parker, A.R. Khoogar and D.E. Goldberg (1989). Inverse kinematics of redundant robots using genetic algorithms. *Proceedings of IEEE International Conference on Robotics and Automation*, **1**, 271–276.

[10] W.M. Yun and Y.G. Xi (1996). Optimum motion planning in joint space for robots using genetic algorithm. *Robotics and Autonomous Systems*, **18**(4), 373–393.

[11] Z.M. Bi and W.J. Zhang (2001). Concurrent optimal design of modular robotic configuration. *Journal of Robotic Systems*. **18**(2), 77–87.

[12] B.S. Krishnaswami (1996). A virtual prototyping environment for robot design using a genetic algorithm. Masters Thesis, The University of Texas at Arlington.

[13] S. Faisal (1998). A virtual prototyping environment for the design of mechanical systems using DADS and genetic algorithms. Masters Thesis, The University of Texas at Arlington.

[14] I-Ming Chen and J.W. Burdick (1995). Determining task optimal modular robot assembly configurations. *Proceedings of IEEE International Conference on Robotics and Automation*, **1**, 132–137.

[15] R. Boudreau and N. Turkkan (1996). Solving the forward kinematics of parallel manipulator with a genetic algorithm. *Journal of Robotic Systems*, **13**(2), 111–125.

[16] M. Zhao, N. Ansari and Edwin S.H. Hou (1994). Mobile manipulator path planning by a genetic algorithm. *Journal of Robotic Systems*, **11**(3), 114–153.

[17] R. Storn (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, **11**(4), 341–359.

[18] K. Price and R. Storn (1997). A simple evolution strategy for fast optimization. *Dr. Dobb's Journal*, **264**, 18–24.

[19] http://www.icsi.berkeley.edu/~storn/code.html#appl, 2001.

[20] D.E. Goldberg (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Longman, Inc., New York.

[21] J.J. Craig (1986). *Introduction to Robotics*. Addison-Wesley Publishing Company Inc., New York.

[22] R.R. Craig (1981). *Structural Dynamics*, pp. 387–392, John Wiley & Sons, Inc., New York.

[23] Z. Michalewicz (1994). *Genetic Algorithms + Data structures = Evolution Programs*. Springer-Verlag, Berlin Heidelberg.

[24] J.A. Vasconcelos, J.A. Ramirez, R.H.C. Takahashi and R.R. Saldanha (2001). Improvements in genetic algorithms. *IEEE Transactions on Magnetics*, **37**(5), 3414–3417.

[25] (2001). *A Bibliography of Differential Evolution Algorithm*. Lappeenranta University of Technology, Lappeenranta, Finland.

[26] R. Storn (1996). On the usage of differential evolution for function optimization. Fuzzy Information Processing Society, NAFIPS, Biennial Conference of the North American, 519–523.