Connection Science, 2015
Vol. 27, No. 1, 22–39, http://dx.doi.org/10.1080/09540091.2014.948385

Global–local population memetic algorithm for solving the forward kinematics of parallel manipulators

Rohitash Chandra[a,b*] and Luc Rolland[c]

[a]*School of Computing, Information and Mathematical Sciences, Faculty of Science, Technology and Environment, University of the South Pacific, Suva, Fiji;* [b]*Artificial Intelligence and Cybernetics Research Group, Software Foundation, Nausori, Fiji;* [c]*Memorial University, Faculty of Engineering, St-John's, Canada*

(*Received 29 March 2014; accepted 15 July 2014*)

Memetic algorithms (MA) are evolutionary computation methods that employ local search to selected individuals of the population. This work presents global–local population MA for solving the forward kinematics of parallel manipulators. A real-coded generation algorithm with features of diversity is used in the global population and an evolutionary algorithm with parent-centric crossover operator which has local search features is used in the local population. The forward kinematics of the 3RPR and 6–6 leg manipulators are examined to test the performance of the proposed method. The results show that the proposed method improves the performance of the real-coded genetic algorithm and can obtain high-quality solutions similar to the previous methods for the 6–6 leg manipulator. The accuracy of the solutions and the optimisation time achieved by the methods in this work motivates for real-time implementation of the 3RPR parallel manipulator.

Keywords: forward kinematics of parallel manipulators; real-coded genetic algorithm; memetic algorithms; intensification and diversification

1. Introduction

Evolutionary algorithms (EAs) are inspired from biological evolution and uses a population of solutions that are evolved over time for a specific goal. The goal is usually minimisation or maximisation of one or several objective functions. EAs are often called *metaheuristic* algorithms (Blum & Roli, 2003).

In robotics, EAs have been applied for solving the forward kinematics problem (FKP). Initial work was done by Boudreau and Turkkan (1995), who used genetic algorithms for solving the FKP of 3RPR manipulators where it was reported that genetic algorithms are more time consuming than Newton–Raphson's method (Ypma, 1995). Newton's method suffers from premature convergence and Jacobian inversion problems. The FKP of spherical and spatial manipulators has been approached with EAs, which are known as global search techniques, that are not restricted by Jacobian inversion problems. EAs can also be used for problems that are non-differentiable.

The robot kinematics problems are mostly associated with solving a system of nonlinear equations that can be expressed as direct optimisation problem where the objective function describes

*Corresponding author. Email: c.rohitash@gmail.com

the entire dynamics of the system (Boudreau, Darenfed, & Turkkan, 1998; Wang, Hao, & Cheng, 2008).

In our previous work, we have used real-coded genetic algorithms (RCGAs) (Rolland & Chandra, 2009b) for solving the forward kinematics of the 3RPR parallel manipulator and 6 leg parallel manipulator (Rolland & Chandra, 2009a). Initially, the solution quality was not as good as algebraic methods (Rolland, 2006). Further attempt was using simulated annealing and hybrid metaheuristic methods where simulated annealing and genetic algorithms were combined (Chandra, Zhang, & Rolland, 2009). These methods achieved better performance than older versions of RCGAs (Rolland & Chandra, 2009b), however, the solution quality was not as good when compared to algebraic methods (Rolland, 2006).

In our recent works, we used the G3-PCX (generalised generation gap with parent-centric crossover (PCX)) EA (Deb, Anand, & Joshi, 2002) for solving the forward kinematics of the 6–6 leg parallel manipulator (Rolland & Chandra, 2010). The G3-PCX EA showed the best results in comparison to other EAs in terms of solution quality and reduced optimisation time for the 6–6 leg problem. It reported all the 16 distinct solutions with very good solution quality (Rolland & Chandra, 2010) when compared to algebraic methods (Rolland, 2007). In this paper, we apply the G3-PCX for the 3RPR problem.

Memetic algorithms (MA) (Moscato, 1989) typically combine population-based EAs with local search that provides intensification. The search for more efficient local refinement techniques has been a major focus of study in MAs. It has been shown that EAs can also be used as effective local search techniques (Kazarlis, Papadakis, Theocharis, & Petridis, 2001; Lozano, Herrera, Krasnogor, & Molina, 2004; Molina, Lozano, García-Martínez, & Herrera, 2010). The use of EAs for local search is known as *crossover-based local search* that is implemented as a *local population*. The local population makes use of efficient crossover operators that have local search properties. The local population-based MA have shown promising results by achieving quality solutions in comparison with other evolutionary approaches for function optimisation problems with high dimensionality (Molina et al., 2010).

This paper presents *Global–Local Population Memetic Algorithm* (GLPMA) for solving the forward kinematics of the 3RPR and the 6–6 leg parallel manipulator. The proposed method employs RCGA with Wrights heuristic crossover (Wright, 1991) and roulette wheel selection in the global population and G3-PCX in the local population in order to balance diversification with intensification. The optimisation performance is compared with the previous methods used for the same problem (Rolland & Chandra, 2009a, 2010). The G3-PCX is used for solving the 3RPR problem for the first time in this paper.

The rest of the paper is organised as follows: Section 2 presents the details of the proposed global–local population MA. In Section 3, the FKP of parallel manipulators and its conversion into an optimisation problem is given in detail. Section 4 presents the results and discussion on the analysis of the results. Section 5 concludes the paper with directions for future research.

2. Background

2.1. *Related work on MAs*

Global search or the process of diversification traverses over several neighbourhoods of solutions while local search limits itself within a single solution neighbourhood. The neighbourhood $N(v)$ of a vertex v is the sub-graph that consists of the vertices adjacent to v (not including v itself) (Watts, 1999). Local search or the process of intensification is also viewed as hill-climbing that refines the solution. Evolutionary search methods begin with global search that contains large difference between candidate solutions in the population. As the search progresses, with evolutionary operators such as selection and recombination, the search points to a single solution neighbourhood and the candidate solutions are closer to each other.

MAs (Moscato, 1989) typically combine EAs with local search in order to provide a global solution of improved quality. The local search is also known as individual learning, local refinement or intensification. The individual that undergoes local search is known as the *meme*. MAs also include the combination of EAs with problem-dependant heuristics and approximate methods and special recombination operators (Moscato, 2003). Applications of MA include difficult combinatorial optimisation problems, machine learning and robotics, molecular optimisation problems, electronics and engineering and other optimisation problems as discussed in Moscato (2003). A review on MAs appears in Ong, Lim, Zhu, and Wong (2006) and a progress report indicates that they are an emerging field in evolutionary computation (Ong, Lim, & Chen, 2010).

A growing field of interest is in using EAs for local search methods in MAs. The EA for local search has a small population size which is evolved for a short duration (Kazarlis et al., 2001; Lozano et al., 2004; Molina et al., 2010; Soak, Lee, Mahalik, & Ahn, 2006). The EA with the large population is known as the master or *population* intended for global search (diversification) and the population for local search (intensification) is known as the *subordinate or local* population. In the rest of the paper, we refer to the EA used for local search as *local population*.

Kazarlis et al. (2001) introduced the concept of micro genetic algorithm for local search whereby a population of few individuals was employed as a generalised hill-climber intended for intensification and a genetic algorithm with a larger population were used for diversification. Lozano et al. (2004) presented a real-coded MA with *crossover hill-climbing* that maintains a pair of parents which consists of the solution being refined and along with the best solution. The crossover operation is performed on the pair until some number of offspring has been reached. The best offspring is selected and it replaces the worst parent only if the best solution is better. The method performed better than the MAs is presented in the literature.

Noman and Iba (2008) incorporated an adaptive crossover hill-climbing method for differential evolution where the intensity of local search is adjusted adaptively. They proposed fixed length and adaptive method for the intensity of local search. In the adaptive method, the crossover hill-climbing method is evolved while the offspring performs better than the first parent. If the performance of the offspring is worse, then the search returns to the differential evolution. Soak et al. (2006) presented an MA which used ideas from particle swarm optimisation for diversification and recombination operators from a genetic algorithm were used for intensification. The method obtained promising results to several instances of the constrained minimum spanning tree problem. Mutoh, Kato, and Itoh (2005) presented the flexible-step crossover operator which performed local search and the results showed improved performance for continuous optimisation problems. Gang, Iimura, Tsurusawa, and Nakayama (2005) used a global genetic algorithm as the master and a local genetic algorithm as the subordinate for the travelling salesman problem.

Molina et al. (2010) used the covariant matrix adaptation evolution strategies (Hansen & Ostermeier, 2001) with a local population. They used a steady-state genetic algorithm (Herrera, Lozano, & Verdegay, 1998) as global population which has the property of high population diversity. The method showed good results for continuous problems with high dimensionality when compared to its counterparts from the literature.

2.2. *Generalised generation gap with parent-centric crossover*

G3-PCX has been applied to 6-leg FKPs in the past (Rolland & Chandra, 2010). It is important to highlight its strengths and limitations as studied in benchmark problems.

In past research, the PCX operator was compared with simplex crossover and simulated-binary crossover using the generalised generation gap model. The results showed that the PCX operator achieved improved performance in terms of lower optimisation time and also scaled up better as the problem size was increased. These simulations were limited to three unimodal functions and further comparisons of G3-PCX with differential evolution (Storn & Price, 1997) and evolution strategies (Hansen & Ostermeier, 2001) showed improved performance in terms of optimisation time and scalability (Deb et al., 2002).

The advantage of the PCX operator is that it behaves like a mutation operator and at the same time retains diversity and is self-adaptive. It has been used as a hill-climbing local search procedure in an MA (Lozano et al., 2004). There is no mutation operator in the original G3-PCX algorithm. Amendments to the original algorithm have been done by proposing a mutation operator (Teo, Hijazi, Omar, Mohamad, & Hamid, 2007) which has shown good performance in multi-modal problems. Further amendments have been done in the PCX operator by introducing a female and male differentiation process which determines the male and female individuals chosen from the population and by further using parent selection mechanisms shown in García-Martínez, Lozano, Herrera, Molina, & Sánchez (2008). A roulette wheel-based parent selection scheme has also shown to perform better than the original G3-PCX on highly nonlinear multi-dimensional problems (Ray, Kok, & Kian, 2004).

The G3-PCX needs fairly large population for small problems (Pošík, 2009). In a study, several short experiments revealed that even for two-dimensional problems, a population size of 90 was needed to find the solution reliably. The population size of 300 was needed in order to solve a 40-dimensional sphere function (Pošík, 2009).

A major limitation of G3-PCX is in multi-modal optimisation problems as shown in Pošík (2009) where a restart scheme also showed to be inappropriate. This problem can be handled if more diversity is given to G3-PCX as PCX has more emphasis for local search.

3. Global–local population MA

As discussed earlier, MAs have employed local search using local populations that use crossover-based local search to balance diversification with intensification. The choice of the particular EA in the global and local population is dependent on the application problem.

The proposed GLPMA employs the same EA in the global and local population. The G3-PCX is the designated evolution algorithm in the local population due to the features of the parent-centric crossover in terms of local search (Deb et al., 2002). The GLPMA is given in Algorithm 1. The RCGA with Wrights heuristic crossover (Wright, 1991) and roulette wheel selection is used in the global population. The algorithm begins by initialising all the individuals of the respective populations with real-random numbers in a given distribution. The populations are evaluated by presenting each individual to the fitness function which defines the FKP of the respective parallel manipulator. Once the populations are evaluated, the algorithm proceeds as a standard EA which employs genetic operators such as selection, crossover and mutation to create new offspring for the master population. The algorithm assumes that it has been given the best parameters for the evolutionary operators such as the crossover and mutation rate. The population size of the respective populations needs to be evaluated to suit the problem.

After certain number of generations of evolution in the global population, the best individual from the global population is transferred to the local population that employs G3-PCX. The *local search intensity* determines how often to apply local search and *local search intensity* determines how long to apply them. Both parameters need to be optimised. They are given by the number of generations in the respective populations and are dependent on the problem type, size in terms

Algorithm 1 GLPMA

Randomly initialise the master population (RCGA)
Evaluate master population
Randomly initialise the local population (G3-PCX)
Evaluate local population

while Not Termination **do**
  (i) Create new individuals using genetic operators (Wrights heuristic crossover)
  (ii) Update master population
  if (LOCAL SEARCH) **then**
    Carry best individual from master population as a meme M
    Local refinement on M for n iterations using local population (PCX)
  end if

  Replace the worst individual of the master population with the refined meme

end while

of dimension and the nature of the search space. When the local search is executed, then the chosen individual from the global population is transferred as a meme and placed into the local population depending local search intensity given by number of generations. The best individual in the global population is often chosen as the meme, however, in some cases, the second- or the third-best individuals can also be chosen.

The meme is then refined using crossover-based local search in the local population. The refined meme is then copied to the global population where it replaces the worst individual. Note that even if the refined meme is not improved, it replaces the worst individual, as it may have features which will be used later in the evolutionary process. Although crossover-based local search is used as the designated algorithm for local refinement, any other local search method can be employed that includes hill-climbing and gradient-based methods. The local population is evolved as specified by the local search intensity in terms of the number of generations. The best individual is then transferred to the main population. The remaining individuals in the local search population are kept and used in future local search evolution. The details of the G3-PCX is discussed in the following section.

3.1. *G3-PCX for local population*

The G3-PCX EA is used for the local population as the parent-centric crossover operator has local search features. The G3-PCX EA is also used for the global population, however, any other evolutionary method can also be used.

In G3-PCX, the whole population is randomly initialised and evaluated as done in the canonical genetic algorithm. The difference lies in the selection method where a small sub-population is made of few chosen parents and children. At each generation, only the sub-population is evaluated rather than evaluating the whole population as in a standard genetic algorithm. The children with their new fitness become part of the bigger population. The best individual of the population is retained at each generation.

The parent-centric crossover operator is used in creating an offspring based on orthogonal distance between the parents. The parents are made of female and male components. The *female*

Algorithm 2 G3-PCX EA (Deb et al., 2002)

Set the number of *parents* (α) and *children* (β)
initialise and evaluate all individuals in the *population*
Setup the *sub-population* which would contain parents and children
while not optimal solution **do**
  (1) Select the best *parent* and $\alpha - 1$ *parents* randomly from *population*
  (2) Create β *children* from α *parents* using PCX
  (3) Choose two parents at random from the *population*
  (4) From the combined *sub-population* of two chosen parents and β created children, choose the two best individuals and replace the chosen two parents (in Step 3) with these solutions.
end while

parent points to search areas and the *male* parent is used to determine the extent of search of the areas pointed by the female. The genes of the offspring extract values from intervals associated with the neighbourhood of the female and male using probability distribution. The range of this probability distribution depends on the distance among the genes of the male and the female parents. The PCX operator assigns more probability to create and offspring near the female than anywhere else in the space. The general procedure used in the G3-PCX is given in Algorithm 2.

Algorithm 2 begins by initialising and evaluating all the individuals in the population. The sub-population is then created which has the size of number of parents and the children. The number of parents and children must be defined beforehand. The best parent is selected from the population and the rest of the parents are selected randomly. The selection of the best parent ensures elitism in the procedure. The children are created from the parents using the PCX crossover operator given in Deb et al. (2002). The parents and the children are combined in the sub-population. Afterwards, *n* strong individuals are chosen from the combined sub-population which are further replaced in the population.

4. Forward kinematics of parallel manipulators

The idea to design planar parallel mechanisms can be traced back as early as in the beginning of the 1940s with Pollard and his *five-bar mechanism*.[1] Based on the *Multi-Axis Simulation Tables*, Gough and Cappel constructed the first parallel hexapods, respectively, for a tire testing device (Gough & Whitehall, 1962) and a motion simulator (USA Patent no. 3,295,224). Since their successful application as flight simulators (Stewart, 1965), parallel manipulators have attracted academic and industrial interest.

The design hypothesis states that the bodies are infinitely rigid and the joints neither yield friction nor play. The redundant manipulators shall not be studied in this article. Moreover, the number of actuated and measured joint variables equals the number of end-effector *degrees-of-freedom*.

This article shall only concentrate on the FKP. Usually, the inverse kinematics problem (IKP) is required to model the FKP and is defined as follows: *given the generalised coordinates of the manipulator end-effector, find the joint positions.* Accordingly, the FKP is defined as follows: *given the joint positions, find the generalised coordinates of the manipulator end-effector.* In the majority of parallel manipulator cases, the FKP is a difficult problem (Raghavan & Roth, 1995).

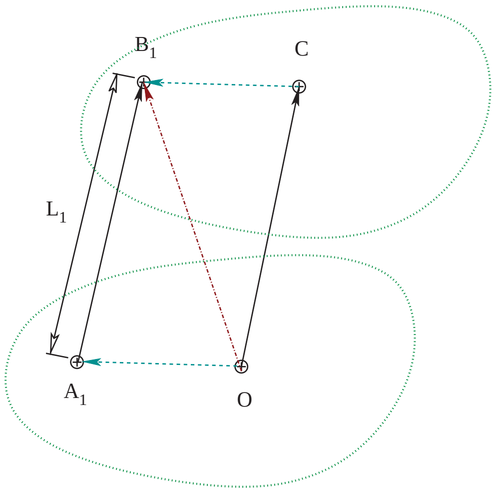

Figure 1. Kinematics chain closed vector cycle.

4.1. *Vectorial formulation of the basic kinematics model*

The vectorial formulation constructs an equation system for each kinematics chain as a closed vector cycle between: (1) the A_i and B_i kinematics chain attachment points, (2) the fixed base reference frame O, and (3) the mobile platform reference frame C (Dieudonne, Parrish, & Bardusch, 1972). An implicit function $\overrightarrow{A_iB_i} = U_1(X)$ can be written between joint positions A_i and B_i for each kinematics chain. Each vector $\overrightarrow{A_iB_i}$ is expressed knowing the joint coordinates $\bar{L}$ and X that give function, $U_2(X, \bar{L})$, $X = (x_c, y_c, \theta)$. The following equality has to be solved: $U_1(X) = U_2(X, \bar{L})$. The distance between A_i and B_i is set to l_i. Thus, the end-effector position X or C can be derived by one platform displacement $\overrightarrow{OC}$ and then one platform general rotation expressed by the rotation matrix $\mathcal{R}$. The vectorial formulation shown in Figure 1 evolves as a displacement-based equation system using the following relation:

$$\overrightarrow{A_iB_i} = \overrightarrow{OC} + \mathcal{R}\overrightarrow{CB_i} - \overrightarrow{OA_i}. \tag{1}$$

For each distinct platform point $\overrightarrow{OB_{iO}}$ with $i = 1, \ldots, 6$, each kinematics chain can be expressed using the distance norm constraint (Merlet, 1997):

$$l_i^2 = \|A_iB_i\|^2. \tag{2}$$

4.2. *The general planar parallel manipulator*

Typically, planar parallel manipulators are characterised by one base, one mobile platform and three kinematics chain which lie in one plane, namely the 3-RPR (Gosselin & Merlet, 1994). Moreover, the manipulator end-effector displacements are restricted to that same plane as shown in Figure 2. A review of planar parallel manipulators shows us that the majority of the proposals fall into the following four classes: 3RPR, 3RRP, 3PRR and 3RRR (Rolland, 2006). We shall only study the 3RPR. For each kinematics chain, the RPR manipulator is constituted by a prismatic actuator located between two ball joints fixed on the base and the platform. Let us have $L_i = l_i$.

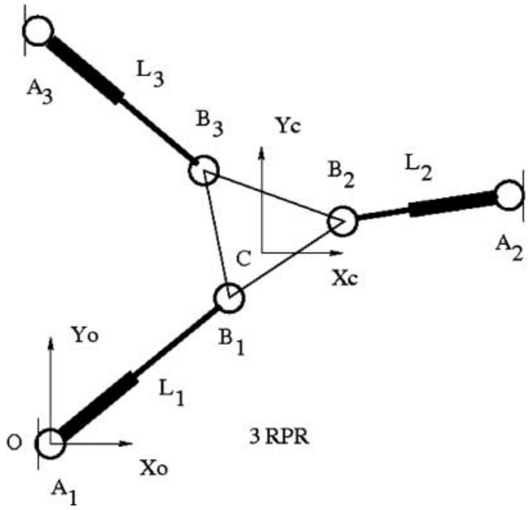

Figure 2. The general planar manipulator and the typical 3-RPR tripod (Rolland, 2006).

The kinematics model is an implicit relation between the configuration parameters and the posture variables, $F(X, \bar{L}OA_{|R_f}, CB_{|R_m}) = 0$, where $\bar{L} = \{l_1, l_2, l_3\}$.

Vectorial formulation in Equation (2) evolves as a displacement-based equation system using the relation given in Equation (1).

$$A_iB_i = OC + \mathcal{R}CB_i - OA_i. \tag{3}$$

As given in Equation (3), for this planar manipulator, the rotation matrix $\mathcal{R}$ is expressed in Equation (4).

$$\mathcal{R} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}, \tag{4}$$

$$L_i^2 = \|A_iB_i\|^2. \tag{5}$$

For each distinct platform point $OB_{i|R_f}$ with $i = 1, \ldots, 3$, each kinematics chain can be expressed using the distance norm constraint (Merlet, 1997).

4.2.1. *The forward kinematics and conversion to optimisation problem*

EAs are optimisation methods whereas the FKP involves solving a system of nonlinear equations. The FKPs need to be converted into an optimisation problem in order to make use of EAs. The inverse kinematic model is required from which we can easily derive an objective function that is also called the fitness function. The fitness function represents the total error on each leg lengths. Let lg_i be the leg length of kinematics chain i which is given as input of the problem. Therefore, the fitness function f, is given in Equation (6).

$$f = \sum_{i=1}^{3} (l_i - lg_i)^2. \tag{6}$$

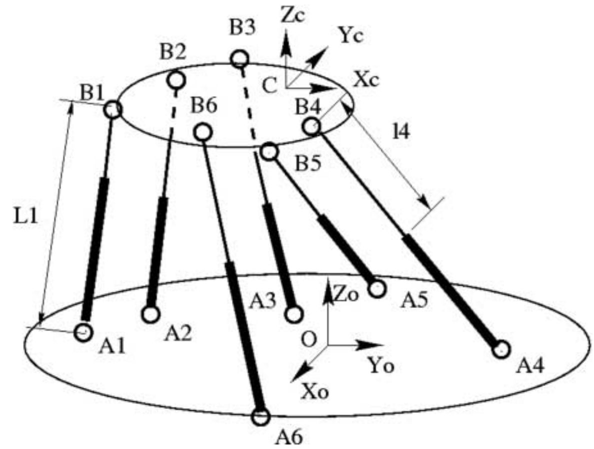

Figure 3. The 6 leg parallel manipulator (Rolland, 2007).

If we set $H_i = l_i^2$, from Equation (6), then the fitness function is updated as given in Equation (7):

$$f = \sum_{i=1}^{3} (\sqrt{(H_i)} - lg_i)^2. \tag{7}$$

4.3. *The 6–6 hexapod parallel manipulator*

In the FKP of 6–6 leg parallel manipulator, shown in Figure 3, the kinematics model is an implicit relation between the configuration parameters and the posture variables, $F(X, \bar{\rho}, OA_{|R_f}, CB_{|R_m}) = 0$, where $\bar{L} = \{l_1, \ldots, l_6\}$.

For each distinct platform point $OB_i|R_f$ with $i = 1, \ldots, 6$, each kinematics chain can be expressed using the distance norm constraint (Merlet, 1997):

$$L_i^2 = \|A_i B_i\|^2. \tag{8}$$

4.3.1. *The inverse kinematics problem*

We shall examine one formulation derived from the position-based equations. Selig demonstrates how three points can be applied to describe the position and displacement of any rigid body (Selig, 1992). It was then applied to the forward kinematics model of parallel manipulators as shown in Figure 4. The coordinates of the three distinct points become the nine variables from which constraints equation can be written. We apply this principle in the positioning of the parallel manipulator mobile platform where the end-effector is fixed rigidly. This position-based model can then give the pose of the parallel manipulator end-effector. There are two reasons that justify this choice. The model does not separate the mobile platform position and orientation. Every variable then have the same units and their range is equivalent leading to same weight. Hence, the rotation impact is included into the point parameters and made equivalent to the translation impact. The main disadvantage is the unknown number exceeding the end-effector degree-of-freedom number (Rolland, 2005).

The three platform distinct points are usually selected as the three joint centres, B_1, B_2, B_3. The nine variables are set as follows: $\overrightarrow{OB}_{i|O} = [x_i, y_i, z_i]$ for $i = 1, 2, 3$. To simplify the computation,

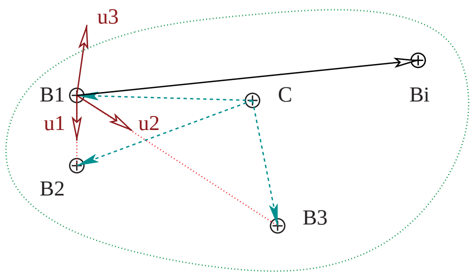

Figure 4. Mobile platform position specification by the three points.

we choose one non-Cartesian reference frame R_{b_1} to be located on B_1. The u_1, u_2 and u_3 frame axes are defined by the following equations:

$$u_1 = \frac{\overrightarrow{B_1B_2}}{\|\overrightarrow{B_1B_2}\|}, \quad u_2 = \frac{\overrightarrow{B_1B_3}}{\|\overrightarrow{B_1B_3}\|}, \quad u_3 = u_1 \wedge u_2. \tag{9}$$

This new reference frame R_{b_1}, substitutes R_m, the mobile platform Cartesian reference frame. This is achieved to produce a simpler equation system. Knowing that the platform is supposed infinitely rigid, any platform point M can be expressed as follows:

$$\overrightarrow{B_1M} = a_M u_1 + b_M u_2 + c_M u_3, \tag{10}$$

where a_M, b_M, c_M are constants in terms of these three points. Hence, in the case of the IKP, the constants are noted $a_{B_i}, b_{B_i}, c_{B_i}$, $i = i \cdots 6$, and can explicitly be deduced from $CB_{|C}$ by solving the following linear system of equations:

$$\overrightarrow{B_1B}_{i|R_{b_1}} = a_{B_i}u_1 + b_{B_i}u_2 + c_{B_i}u_3, \quad i = 1 \cdots 6. \tag{11}$$

Using the relations in Equation (11), the distance constraint equations $l_i^2 = \|\overrightarrow{A_iB}_{i|O}\|^2$, $i = 1, \ldots, 6$ can be expressed. Thus, for $i = 1, \ldots, 6$, the IKP is obtained by isolating the l_i actuator variables in the six following equations:

$$l_i^2 = (x_i - OA_{ix})^2 + (y_i - OA_{iy})^2 + (z_i - OA_{iz})^2, \quad i = 1, \ldots, 3 \tag{12}$$

$$l_i^2 = \|\overrightarrow{B_i}_{|R_{b_1}} - \overrightarrow{OA}_{iO}\|^2, \quad i = 4, \ldots, 6. \tag{13}$$

4.3.2. *The forward kinematics problem*

The IKP expression gives an algebraic system comprising the first six equations in terms of three point variables: $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$, Equation (13).

The system of nonlinear equations in the FKP is converted to an optimisation problem as done for the 3RPR in the previous section. Hence, from the IKP, we can easily derive a fitness function. This function will be calculated on each FKP estimation that represents the total error on each kinematics chain lengths. Let lg_i be the length of kinematics chain i which is given as problem input. Let $H_i = l_i^2$, from Equation (13), the fitness function becomes:

$$f = \sum_{i=1}^{6} (\sqrt{(H_i)} - lg_i)^2. \tag{14}$$

Equation (14) contains six individual objectives being the kinematics chain length difference constraints which will be minimised.

4.3.3. *Extended fitness function*

Preliminary tests led to several solutions which were absolutely not in correspondence with the exact proven ones (Chandra, Frean, & Rolland, 2009). In other words, the selected fitness function was incorrect since it did find other solutions or solutions which were not the results of solving the *FKP*.

To alleviate this problem, the selected fitness function is then augmented by the three constant distances between the three distinct platform points, B_1, B_2 and B_3, given as follows.

$$\begin{aligned}
G_1 &= \|\overrightarrow{B_2 B_{1C}}\|^2 - (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2, \\
G_2 &= \|\overrightarrow{B_3 B_{1C}}\|^2 - (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2, \\
G_3 &= \|\overrightarrow{B_3 B_{2C}}\|^2 - (x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2.
\end{aligned} \tag{15}$$

Hence, the fitness function is given in the following equation.

$$f = \sum_{k=1}^{3} (\sqrt{H_k} - kg_k)^2 + \sum_{n=1}^{3} (G_n)^2. \tag{16}$$

The fitness function includes the nine variables obtained from the six kinematics chain lengths and three platform distance constraints.

5. Experiments and results

In this section, the performance of the proposed GLPMA and G3-PCX EA is compared for the given FKPs as discussed in Section 4. The comparison is made with the G3-PCX EA (Rolland & Chandra, 2010) which has given the best performance when compared to hybrid genetic algorithms – simulated annealing and RCGAs (Rolland & Chandra, 2009a).

The G3-PCX will be applied for the 3RPR problem for the first time in this paper. We choose the G3-PCX because it is fast, robust and requires fewer parameters to be optimised. The G3-PCX algorithm does not use a mutation operator and its crossover operator is based on the parent-centric mechanism which also has mutation like features and has local search properties. Hence, it is used as the designated algorithm for the local population in the GLPMA.

The following steps are taken in this section to test the proposed method on the FKP problem for 3RPR and 6–6 leg manipulator, respectively.

- Step 1: Evaluate the frequency and local search intensity in the GLPMA.
- Step 2: Compare GLPMA, RCGA and G3-PCX EA.

The G3-PCX algorithm uses a pool size of two kids and a family size of two parents for all the respective population sizes. We use the original G3-PCX given in Deb et al. (2002) for the FKP problem.

In all experimental set-up, a total of 100 experiments were executed with different random initial populations. The search terminated when the algorithm reached maximum training time or when the fitness value gets lower than 1E−20.

The success rate shows how well the given paradigm can guarantee a solution when given any initial random solution within the search space. The optimisation time is given in seconds(s) in all the experiments.

5.1. *FKP configuration*

5.1.1. *3RPR configuration*

The fitness function is derived from the inverse kinematics of tripod 3-RPR parallel manipulator shown in Equation (7). An example of the resolution on a typical 3RPR manipulator configuration is examined for simulation. The manipulator base coordinates of the joint centre position $OA_{|R_f}$ in the base reference frame R_f and the mobile platform coordinates of the joint centre position $CB_{|R_m}$ in the platform reference frame R_m, and the minimum bar lengths are given in Table 1.

The joint variables are the kinematics chain lengths. The following leg lengths were used: $L := [100, 120, 150]$. All experiments initialise the population with real numbers in the range of $[-50, 50]$. The variables that need to be optimised are the position's x, y and θ and therefore, this is a three-dimensional problem. An experimental run was considered successful when the algorithm terminated by obtaining an error of 1E−20 before the maximum training time is reached.

5.1.2. *The 6–6 hexapod configuration*

We examine one FKP example on a typical 6–6 parallel manipulator configuration, Table 2 shows the fixed base and mobile platform joint coordinates, $PA_{|O}, CB_{|C}$.

Table 1. Planar parallel manipulator configuration for 3RPR.

Config.	$A_1(x)$	$A_1(y)$	$A_2(x)$	$A_2(y)$	$A_3(x)$	$A_3(y)$
3RPR	0	0	200	0	0	200
Id	$B_1(x)$	$B_1(y)$	$B_2(x)$	$B_2(y)$	$B_3(x)$	$B_3(y)$
3RPR	0	0	50	0	40	40

Table 2. Parallel manipulator configuration table.

Joint coordinates			Respective values		
$OA_1(x)$	$OA_1(y)$	$OA_1(z)$	464.141	389.512	−178.804
$OA_2(x)$	$OA_2(y)$	$OA_2(z)$	569.471	207.131	−178.791
$OA_3(x)$	$OA_3(y)$	$OA_3(z)$	529.050	−597.151	−178.741
$CB_1(x)$	$CB_1(y)$	$CB_1(z)$	68.410	393.588	236.459
$CB_2(x)$	$CB_2(y)$	$CB_2(z)$	375.094	−137.623	236.456
$CB_3(x)$	$CB_3(y)$	$CB_3(z)$	306.664	−256.012	236.461

The prismatic actuator variables are set, respectively, to $\bar{L} = [1250, 1250, 1250, 1250, 1250, 1250]$. We have deliberately chosen 1 difficult case with 16 exact real solutions. The fitness function derived from the inverse kinematics of tripod 6–6 leg parallel manipulator, given in Equation (16), is used.

All experiments initialise solutions with real numbers in the range of $[-1000, 1000]$. This range is chosen to provide the genetic algorithm with a greater search space since several distinct solutions are present. The solutions contain nine real variables which represent the positions x_i, y_i and z_i, where $i = 1, 2$ and 3. An experimental run was considered successful when the algorithm terminated by obtaining an error of 1E−20 before the maximum training time is reached.

5.2. *Experiments and results for the FKP of the 3RPR*

In this section, the FKP problem of the 3RPR parallel manipulator is examined.The maximum evolution time is specified by 10,000 function evaluations. The algorithm converges if the fitness reaches the minimum error of 1E−20. A run is considered successful if it converges to the minimum error before reaching the maximum number of function evaluations. The success rate shows how well the given paradigm can guarantee a solution when given any initial random solution within the search space. In the GLPMA, population size of 100 is used in the global and local search population, respectively. These values have been obtained from trial experiments.

Table 3 evaluates the LS-interval for the FKP of the 6 leg parallel manipulator given a fixed LS-intensity of five generations. The results show that the LS-interval of 5, 7 and 9 gives the lowest values for the optimisation time in terms of function evaluations. The optimisation time increases as the LS-interval increases to larger values. LS-interval of 1 also does not give good performance.

Table 4 evaluates the LS-intensity given a fixed LS-interval of 5 obtained from results in Table 3. The results show that the LS-intensity of 1–11 gives the lowest values for the optimisation time, however, the best results are given with the LS-intensity of 1 when we consider both the number of function evaluations and elapsed time in seconds. The optimisation time did not have major effect as the LS-intensity increases for this problem.

The comparison of the results of the GLPMA with the G3-PCX in Table 5 shows that the proposed method has not given much improvements of the results for the FKP of the 3RPR parallel manipulator when compared to RCGA. The RCGA in this case shows one of the best performance. If the same population size is used in G3-PCX, then there is local convergence. G3-PCX requires a large population size even for small problems such as the 3RPR where only three variables are optimised.

Table 3. An evaluation of LS-interval for the 3RPR (LSI = 5).

LS-interval	Func. Eval	Time (s)
1	8356 $_{253}$	0.071 $_{0.003}$
3	7701 $_{294}$	0.065 $_{0.003}$
5	7605 $_{206}$	0.065 $_{0.002}$
7	7602 $_{236}$	0.067 $_{0.002}$
9	7561 $_{274}$	0.065 $_{0.002}$
11	7817 $_{284}$	0.068 $_{0.003}$
13	7698 $_{236}$	0.068 $_{0.002}$
15	7679 $_{264}$	0.067 $_{0.003}$
17	7768 $_{252}$	0.068 $_{0.002}$
19	8034 $_{233}$	0.071 $_{0.002}$

Table 4. An evaluation of LS-intensity for the 3RPR (LSF = 5).

LS-intensity	Func. Eval	Time (s)
1	7448_{243}	$0.059_{0.002}$
6	7568_{223}	$0.065_{0.002}$
11	7848_{281}	$0.069_{0.003}$
16	8232_{294}	$0.072_{0.003}$
21	8120_{265}	$0.067_{0.002}$
26	8403_{198}	$0.068_{0.002}$
31	8737_{292}	$0.074_{0.003}$
36	8676_{215}	$0.065_{0.002}$
41	8647_{255}	$0.062_{0.002}$
46	9025_{243}	$0.066_{0.003}$
51	8927_{240}	$0.064_{0.003}$
56	8844_{224}	$0.067_{0.003}$

Table 5. Comparison of different methods for the 3RPR.

Method	Func. Eval	Time (s)	Pop. size	Success rate
RCGA	7301_{218}	$0.056_{0.002}$	50	100
RCGA	5111_{189}	$0.029_{0.001}$	100	100
G3-PCX	2984_{73}	$0.012_{0.002}$	100	90
GLPMA	7448_{243}	$0.059_{0.002}$	(100,100)	100

Table 6. An evaluation of LS-interval for the 6-6 leg.

LS-interval	Func. Eval	Time (s)
1	$237{,}769_{547}$	$2.130_{0.116}$
3	$221{,}697_{697}$	$2.447_{0.103}$
5	$228{,}401_{797}$	$2.693_{0.172}$
7	$239{,}761_{597}$	$2.901_{0.138}$
9	$245{,}161_{1209}$	$3.025_{0.159}$
11	$251{,}777_{490}$	$3.152_{0.152}$
13	$266{,}569_{697}$	$3.347_{0.141}$
15	$270{,}545_{797}$	$3.424_{0.127}$
17	$271{,}409_{793}$	$3.441_{0.130}$
19	$285{,}441_{914}$	$3.638_{0.153}$

5.3. *Experiments and results for the FKP of the 6–6 leg*

In this section, the FKP problem of the 6–6 leg parallel manipulator is examined. The maximum evolution time is specified by 500,000 fitness evaluations. The algorithm converges if the fitness reaches the minimum error of 1E−20. In the GLPMA, the global and local population size of 200 is used. These values have been obtained from trial experiments.

Table 6 evaluates the LS-interval given a fixed LS-intensity of five generations. The results show that the LS-interval of 1–7 gives the lowest values for the optimisation time with the best elapsed time in seconds. Afterwards, the results deteriorate as the LS-interval increases.

Table 7 evaluates the LS-intensity for the FKP of the 6 leg parallel manipulator given a fixed LS-interval of five generations. The results show that the LS-intensity of 85 gives the lowest values for the optimisation time with the best elapsed time in seconds. The optimisation time increases as the LS-interval increases.

The comparison of the results of the GLPMA with the G3-PCX in Table 8 shows that it has improved the results for the FKP of the 6–6 leg parallel manipulator when compared to RCGA

Table 7. An evaluation of LS-intensity for the 6–6 leg.

LS-intensity	Func. Eval	Time (s)
5	303,321 $_{1497}$	5.0144 $_{0.231}$
15	258,562 $_{1097}$	3.9586 $_{0.195}$
25	252,011 $_{1197}$	3.5728 $_{0.186}$
35	235,009 $_{60}$	3.1322 $_{0.165}$
45	232,169 $_{1097}$	3.039 $_{0.231}$
55	225,064 $_{1075}$	2.8014 $_{0.149}$
65	213,216 $_{1105}$	2.673 $_{0.159}$
75	220,635 $_{1097}$	2.5888 $_{0.177}$
85	192,998 $_{597}$	1.9916 $_{0.123}$
95	194,135 $_{1724}$	2.0708 $_{0.132}$

Table 8. Comparison of different methods for the 6-6 leg.

Method	Func. Eval	Time (s)	Pop. size	Success rate
RCGA	302,348 $_{1432}$	3.928 $_{0.136}$	200	100
G3-PCX	80,306 $_{454}$	0.267 $_{0.015}$	200	90
G3-PCX	77,393 $_{1713}$	0.326 $_{0.013}$	600	100
G3-PCX	100,682 $_{2343}$	0.535 $_{0.017}$	1000	100
GLPMA	192,998 $_{597}$	1.9916 $_{0.123}$	(200, 200)	100

alone. GLPMA has not outperformed the G3-PCX which gives the best performance given that the population size is larger. In the case where the G3-PCX employs the same size population size, the success rate is lower due to premature convergence. Therefore, we note that the GLPMA improved the performance of RCGA and G3-PCX when the population size is the same.

We also note that all the methods were successful in finding the several distinct solutions in multiple runs. In the 3RPR problem, we found the two distinct solutions and in the 6–6 leg problem, we found all the 16 solutions as done earlier (Rolland & Chandra, 2010). This was done by running multiple experiments with different initial positions in the search space. In this way, the EA converges towards the solution nearest to the initial search position.

5.4. *Discussion*

The use of RCGA and GLPMA showed very good performance for solving the 3RPR FKP problem in terms of accuracy of the solutions and the optimisation time. The optimisation time given by RCGA is 29 ms with 100% success rate in a population of 50 individuals. G3-PCX gives better results provided that the population size is larger than 100, otherwise, the success rate is not good enough for real work implementation.

The accuracy and optimisation time gives the motivation for real-time implementation of the 3RPR problem which has not been possible with EAs in past work.

In the 3RPR, the GLPMA has not shown a significant improvement when compared to RCGA and G3-PCX. Local search is problem dependent, and this suggests that for the 3RPR problem, G3-PCX and RCGA alone can well balance diversification with intensification. However, the GLPMA and G3-PCX achieves the same solutions in similar optimisation time. Moreover, the transfer of the meme to the local population also takes some time.

In the 6–6 leg problem, the results also show that the GLPMA has given better performance when compared to the G3-PCX and RCGA alone with the same population size. G3-PCX shows the best results when the population is larger. However, the premature convergence (success rate) of G3-PCX can be improved by the proposed GLMPA which motivates further research in other real-parameter optimisation problems. The G3-PCX has limitations for multi-modal problems and requires a fairly large population size due to the properties of PCX (Pošík, 2009). The GLMPA can address some of its shortcomings.

The 6–6 hexapod is a more difficult and larger problem when compared to the 3RPR. The proposed MA did not show much improvement for the 3RPR problem due to the size and nature of the problem. In the 6–6 hexapod, the proposed MA showed improvements in the RCGA through the local population (G3-PCX). This shows that the proposed algorithm is more applicable to larger and more difficult robot kinematics problems where the search landscape requires a greater balance between diversification and intensification.

The GLPMA takes more time when compared to the G3-PCX alone as there is some time taken in the transfer of the memes depending on the local search interval.

The solutions given by the exact algebraic method in the literature are similar to that of the G3-PCX and GLPMA as it achieves a solution quality of 1E−20 with high success rate. The results also show that the local search intensity is an important parameter of GLPMA and its depth on search is dependent on the nature of the problem. The major limitation of the GLPMA framework is the computational cost required in parameter setting; i.e. optimal values for the frequency and the local search intensity. The GLPMA framework is general and other EAs can be used in future for the global and local population to improve the current results.

6. Conclusion

The main problem in building a memetic framework is to balance the diversification with intensification. This work has efficiently utilised a local population with crossover-based local search in the GLPMA. The frequency and local search intensity have shown to be the main attributes that affect the performance of the proposed MA.

The results show that the proposed MA improved the performance of the RCGA for the 6–6 hexapod which is the larger and more difficult problem. Therefore, the proposed method is more applicable to large and difficult robot kinematics problems and can be applied to improve the performance of where EAs similar to the given RCGA have been used. The accuracy of the solutions and the optimisation time achieved by the methods in this work motivates for real-time implementation of the 3RPR parallel manipulator with evolutionary and MAs for control. The results for the 6–6 leg parallel manipulator can be improved further in future research for real-time implementation.

EAs are easier to implement, independent of the problem domain and are not prone to premature convergence. The MA has improved the canonical G3-PCX for the bigger problem (6–6 leg) in terms of success rate which motivates further research in other real-parameter optimisation problems. In future work, MA based methods could be applied to other kinematics problems.

Note

1. W-L-G. Pollard, Spray painting machine, 26 August 1940, USA Patent no. 2,213,108, Evanston, ILL, USA

References

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys, 35*, 268–308.

Boudreau, R., Darenfed, S., & Turkkan, N. (1998). Etude comparative de trois nouvelles approches pour la solution du problème géométrique direct des manipulateurs parallèles. *Mechanism and Machine Theory, 33*(5), 463–477.

Boudreau, R., & Turkkan, N. (1995). Solving the forward kinematics of parallel manipulators with a genetic algorithm. *Journal of Robotics Systems, 13*(2), 111–125.

Chandra, R., Frean, M. R., & Rolland, L. (2009). *A meta-heuristic paradigm for solving the forward kinematics of 6-6 general parallel manipulator*. Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA 2009, 15–18 December 2009, Daejeon, Korea, pp. 171–176.

Chandra, R., Zhang, M., & Rolland, L. (2009). *Solving the forward kinematics of the 3RPR planar parallel manipulator using a hybrid meta-heuristic paradigm*. Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA, pp. 177–182.

Deb, K., Anand, A., & Joshi, D. (2002). A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolution Computing, 10*(4), 371–395.

Dieudonne, J. E., Parrish, R. V., & Bardusch, R. E. (1972). *An actuator extension transformation for a motion simulator and an inverse transformation applying Newton–Raphson's method* (Tech. rep., D-7067). NASA.

Gang, P., Iimura, I., Tsurusawa, H., & Nakayama, S. (2005). Genetic local search based on genetic recombination: A case for traveling salesman problem. In K.-M. Liew, H. Shen, S. See, W. Cai, P. Fan, & S. Horiguchi (Eds.), *Parallel and distributed computing: Applications and technologies* (pp. 202–212). Vol. 3320 of Lecture Notes in Computer Science. Berlin, Heidelberg: Springer.

García-Martínez, C., Lozano, M., Herrera, F., Molina, D., & Sánchez, A. (2008). Global and local real-coded genetic algorithms based on parent-centric crossover operators. *European Journal of Operational Research, 185*(3), 1088–1113.

Gosselin, C., & Merlet, J. P. (1994). The direct kinematics of planar parallel manipulators : Special architectures and number of solutions. *Mechanism and Machine Theory, 29*, 1083–1097.

Gough, V. E., & Whitehall, S. G. (1962, May). *Universal tyre test machine*. Proceedings of the FISITA Ninth International Technical Congress FISITA, London, pp. 117–135.

Hansen, N., & Ostermeier, A. (2001). Completely derandomized self-adaptation in evolution strategies. *Evolution Computing, 9*(2), 159–195.

Herrera, F., Lozano, M., & Verdegay, J. (1998). Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial Intelligence Review, 12*, 265–319. doi:10.1023/A:1006504901164

Kazarlis, S. A., Papadakis, S. E., Theocharis, I. B., & Petridis, V. (2001). Microgenetic algorithms as generalized hill-climbing operators for ga optimization. *IEEE Transactions on Evolutionary Computation, 5*(3), 204–217.

Lozano, M., Herrera, F., Krasnogor, N., & Molina, D. (2004). Real-coded memetic algorithms with crossover hill-climbing. *Evolutionary Computation, 12*(3), 273–302.

Merlet, J. P. (1997). *Traité des nouvelles technologies – Robotique: Les Robots parall'eles* (2nd ed.). Retrieved from http://www.eyrolles.com/Sciences/Livre/les-robots-paralleles-9782866015992

Molina, D., Lozano, M., García-Martínez, C., & Herrera, F. (2010). Memetic algorithms for continuous optimisation based on local search chains. *Evolutionary Computation, 18*(1), 27–63.

Moscato, P. (1989). *On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms* (Tech. Rep.).

Moscato, P. (2003). A gentle introduction to memetic algorithms. In F. Glover & G. A. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 105–144). New York: Kluwer Academic Publishers.

Mutoh, A., Kato, S., & Itoh, I. (2005). *Efficient real-coded genetic algorithms with flexible-step crossover*. The 2005 IEEE Congress on Evolutionary Computation, Vol. 2, pp. 1470 – 1476.

Noman, N., & Iba, H. (2008). Accelerating differential evolution using an adaptive local search. *IEEE Transactions on Evolutionary Computation, 12*(1), 107–125.

Ong, Y.-S., Lim, M.-H., & Chen, X. (2010). Memetic computation – past, present, future [research frontier]. *IEEE Computational Intelligence Magazine, 5*(2), 24–31. doi:10.1109/MCI.2010.936309

Ong, Y.-S., Lim, M.-H., Zhu, N., & Wong, K.-W. (2006). Classification of adaptive memetic algorithms: A comparative study. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 36*(1), 141–152. doi:10.1109/TSMCB.2005.856143

Pošík, P. (2009). *Bbob-benchmarking the generalized generation gap model with parent centric crossover*. Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, GECCO '09. New York, NY: ACM, pp. 2321–2328. doi:10.1145/1570256.1570324

Raghavan, M., & Roth, B. (1995). Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators. *Transactions of the ASME, 117*, 71–79.

Ray, T. V., Kok, S. W. N., & Kian, P. C. (2004). *Study on the behaviour and implementation of parent centric crossover within the generalized generation gap model*. IEEE Congress on Evolutionary Computation, pp. 1996–2003.

Rolland, L. (2005). Certified solving of the forward kinematics problem with an exact method for the general parallel manipulator. *Advanced Robotics, 19*(9), 995–1025.

Rolland, L. (2006). Synthesis on the forward kinematics problem algebraic modeling for the planar parallel manipulator. *Advanced Robotics, 20*(9), 1035–1065.

Rolland, L. (2007). Synthesis of the forward kinematics problem algebraic modeling for the general parallel manipulator: Displacement-based equations. *Advanced Robotics, 21*(9), 1071–1092.

Rolland, L., & Chandra, R. (2009a). *Forward kinematics of the 6–6 general parallel manipulator using real coded genetic algorithms*. IEEE/ASME Conference on Advanced Intelligent Mechatronics (AIM 2009), pp. 1637–1642.

Rolland, L., & Chandra, R. (2009b). *Forward kinematics of the 3RPR planar parallel manipulators using real coded genetic algorithms*. 2009 24th International Symposium on Computer and Information Sciences, ISCIS 2009, pp. 381–386.

Rolland, L., & Chandra, R. (2010). On solving the forward kinematics of the 6–6 general parallel manipulator with an efficient evolutionary algorithm. In G. Maier, F. G. Rammerstorfer, J. Salenon, V. Parenti Castelli & W. Schiehlen (Eds.), *ROMANSY 18 Robot design, dynamics and control* (pp. 117–124). Vol. 524 of CISM International Centre for Mechanical Sciences. Vienna: Springer.

Selig, J. M. (1992). *Introductory Robotics*. Hemel Hemstead: Prentice-Hall.

Soak, S.-M., Lee, S.-W., Mahalik, N., & Ahn, B.-H. (2006). A new memetic algorithm using particle swarm optimization and genetic algorithm. In B. Gabrys, R. Howlett, & L. Jain (Eds.), *Knowledge-based intelligent information and engineering systems* (pp. 122–129). Vol. 4251 of Lecture Notes in Computer Science, Berlin/Heidelberg: Springer.

Stewart, D. A. (1965) Platform with 6 degrees of freedom. *Proceedings of IMechE, 180*(9), 371–386.

Storn, R., & Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341–359. doi:10.1023/A:1008202821328

Teo, J., Hijazi, H. A., Omar, Z. A., Mohamad, N. R., & Hamid, Y. (2007). *Harnessing mutational diversity at multiple levels for improving optimization accuracy in g3-pcx*. IEEE Congress on Evolutionary Computation, Singapore, pp. 4502–4507.

Wang, X.-S., Hao, M.-L., & Cheng, Y.-H. (2008). On the use of differential evolution for forward kinematics of parallel manipulators. *Applied Mathematics and Computation, 205*(2), 760–769.

Watts, D. J. (1999). *Small worlds: The dynamics of networks between order and randomness*. Princeton, NJ: Princeton University Press.

Wright, A. H. (1991). *Genetic algorithms for real parameter optimization*. Foundations of genetic algorithms (pp. 205–218). Bloomington, IN: Morgan Kaufmann.

Ypma, T. J. (1995). Historical development of the Newton–Raphson method. *SIAM Review, 37*(4), 531–551. doi:http://dx.doi.org/10.1137/1037125