For fixed T this is equivalent to sampling a Markov chain using the Metropolis Monte Carlo method [17], which at equilibrium will visit state S with probability proportional to $E(S)/T$. While a solution with lower energy state than the current solution will always be accepted, solutions with higher energy states may also occasionally be accepted. In optimisation, this has the effect of allowing the algorithm to escape from a local minimum whereas rejecting all steps to higher energy (lower "fitness") states leads only to local minima. The key difference between SA and Metropolis sampling is that, over successive Markov chains, T is slowly decreased toward zero. At high temperature most transitions are accepted. As T decreases, so does the probability of transitioning to a worse solution. Essentially, the algorithm behaves as a randomisation procedure in the initial phase, and turns smoothly into a purely greedy algorithm in the final phase.

The initial temperature, the rate of cooling, and the choice of proposal distribution are up to the user, and may strongly affect performance. A common choice for the proposal distribution is to generate the new solution by simply adding a small real random number [-b,b] to each particle in the new solution. The stopping criteria of the algorithm is usually when ΔE is consistently less than some threshold, or when a minimum temperature is reached, or when the algorithm reaches the maximum number of iterations.

The SA algorithm has been successfully applied to NP hard problems such as traveling salesman problem [16], graph partitioning [18] and VLSI design [19]. It has also been used for training neural networks [20], [21].

B. Genetic Algorithms Overview

Over the past years, the original genetic algorithm(GA) introduced by Holland [22] has evolved significantly in order to suit the real-world optimisation challenges faced by engineers and scientists. One major alteration is in the development of real coded genetic algorithms (RCGA), which use real numbers in their chromosomes rather than bits. The major advantages of these over standard binary-coded GAs are that they maintain precision usually lost in the binary representation of a real number, and require smaller chromosomes, reducing the computation time. Real-valued encoding is also a more natural presentation to use for many real world applications. The computational and optimisation power of RCGA has also been demonstrated in several theoretical studies [23], [24], [25].

In other respects RCGA optimisation is the same as in Holland's original scheme: a number of possible random solutions (chromosomes) make up a population. Over time, each chromosome is evaluated according to its "fitness" and employs "genetic" operators like *selection, crossover* and *mutation* for producing new solutions. These new solutions or "offspring" are added into the population while its least fit members are removed, and the process is repeated. In the standard procedure, two or more parents are selected using the selection operator from a population to make some number of children.

The choice of the appropriate genetic operator is important as it directly influences the convergence of the GA. In practice, different forms of the main genetic operators are needed according to the type of the GA and the nature of the optimisation problem.

C. Hybrid Meta-heuristic Techniques

Hyper-heuristic [26], [27], Hybrid meta-heuristic [28], and Memetic algorithms [29], [30] refer to algorithms where two or more heuristic search techniques are combined to solve difficult problems. This is done to provide intensification and diversification, respectively, at different points during the search process. An example of this approach is the *CoSearch* method proposed by Talbi and Bachelet [31] which uses a combination of genetic algorithms, Tabu search and local search, together with an adaptive memory that contains history of the search already done.

Hybrid meta-heuristic algorithms are grouped into two major categories. In the first category, two or three heuristic search algorithms are combined into a hybrid parallel search paradigm to provide intensification and diversification in search. Examples include the combination of GAs with Tabu search [32] and with simulated annealing [33]. In the second category, one algorithm is used to provide seed points for subsequent search by another algorithm [34], [35]. Both hybrid methodologies are discussed in the following subsections.

The Hybrid GA-SA approach combines both algorithms in parallel where the GA uses the SA for a predefined number of iterations during the GA search process. In the past, the development of hybrid genetic algorithm and simulated annealing has been implemented using parallel processing machine where simulated annealing is employed with lower temperatures after the use of genetic recombination operators such as crossover and mutation over the whole population[36].

This work uses the hybrid approach where the GA uses the SA after the crossover operation (hybrid GA-SA). In this approach, the SA replaces the mutation operator. The hybrid paradigm given in Algorithm(1). Note that the number of iterations (*N*) done during simulated annealing depends on the nature of the problem.

The inner loop of the above algorithm begins by calculating the fitness of each solution in the population. Furthermore, the selection operator chooses two parents depending on the selection criteria. In this implementation, roulette wheel selection is used. The crossover operator combines both parents and produces a single child. The child is presented as a starting point for the simulated annealing algorithm, which is run for *N* iterations over which T is reduced from T_0. The resulting solution from SA is then copied back to the new population.

The simulated annealing process is dependent on a predefined probability which is similar of crossover or mutation probability, and therefore there will be cases where the procedure will not be applied to some members of the population. This implies that weaker solutions are sometimes

Alg. 1 Hybrid GA-SA

Initialize Population (P) of P solutions
while $!termination$ **do**
  while $i < P.size()$ **do**
    1) Evaluate fitness
    2) Selection of *Parent 1* and *Parent 2*
    3) Employ Crossover and produce a single Child
    4) Present the Child to SA for *N* Iter.
    5) Copy the *updated Child* given by SA into *population*
    6) Lower the initial temperature T
    7) increment i
  end while
  Update population
end while
Get the best solution

retained in the future populations as they may contain useful properties for future convergence. This follows the same motivation as in the use of crossover and mutation probability for a standard genetic algorithm.

We used Wright's heuristic crossover operator, as it has shown superior performance in comparison with other crossover operators for a set of optimisation problems [37], [38]. This produces a single offspring given two parents. For a pair of parents $\mathbf{x} = (x_1, x_2, x_3, ...x_n)$ and $x' = (x'_1, x'_2, x'_3, ...x'_n)$, an offspring $\mathbf{y}$is produced as shown in Equation(1):

$$y_i = r(x_i - x'_i) + x_i \tag{1}$$

where r is a real random number in [0,1] and x_i is the parent with the higher fitness. Usually the process is repeated (with independent r) until a chromosome with a better fitness is created.

The *Pivot Mutation* operator introduced in [13] selects a pivot point in the chromosome and all the genes after the selected pivot are mutated by adding different small real-random numbers, respectively. For instance, given a chromosome $\mathbf{x} = (x_1, x_2, x_3, ...x_n)$, the resulting pivoted chromosome becomes $\mathbf{y} = (x_1, x_2, y_3, ...y_n)$, where ,$y_i = x_i + r$, where r is a small real random number in the interval $[a, b]$, where a and b are small negative and positive real numbers chosen by the user, respectively.

III. The Forward Kinematics Problem Formulation

A. The kinematics of parallel manipulators

Any manipulator is characterised by its mechanical configuration parameters and the posture variables. The configuration parameters are therefore; $OA_{|_{R_f}}$, the base attachment point coordinates in R_f (the base reference frame), and $CB_{|_{R_m}}$, the mobile platform attachment point coordinates in R_m (the mobile platform reference frame). The kinematics model variables are the joint coordinates and end-effector generalised coordinates. The joint variables are described as l_i, the prismatic joint or linear actuator positions. The generalised coordinates are expressed as $\overrightarrow{X}$, the end-effector position and orientation.

The kinematics model is an implicit relation between the configuration parameters and the posture variables, $F(\overrightarrow{X}, \overline{\rho}, OA_{|_{R_f}}, CB_{|_{R_m}}) = 0$ where $\overline{L} = \{l_1, \ldots, l_6\}$.

This article shall only concentrate on the forward kinematics problem (FKP) of 6-6 leg parallel manipulator as shown in Fig. 1. Usually the inverse kinematics problem is required to model the FKP and is defined as: *given the generalised coordinates of the manipulator end-effector, find the joint positions.*

Accordingly, the FKP is defined as: *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, [39].

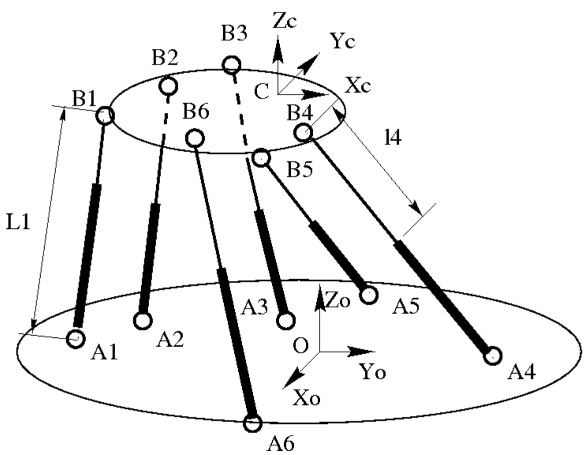

Fig. 1. The 6 leg parallel manipulator [40]

B. Vectorial Formulation of the Basic Kinematics Model

Containing as many equations as variables, vectorial formulation constructs an equation system, [41], as a closed vector cycle between the following points: A_i and B_i, kinematics chain attachment points; O, the fixed base reference frame and C, the mobile platform reference frame. For each kinematics chain, an implicit function $\overrightarrow{A_iB_i} = U_1(X)$ can be written between joint positions A_i and B_i. Each vector $\overrightarrow{A_iB_i}$ is expressed knowing the joint coordinates $\overrightarrow{L_i}$ and X giving function $U_2(X, \overrightarrow{L})$. The following equality has to be solved: $U_1(X) = U_2(X, \overrightarrow{L})$. The distance between A_i and B_i is set to L_i. Therefore, 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 $\mathcal{R}$. Vectorial formulation given in Equation(3) evolves as a displacement based equation system using the following relation :

$$\overrightarrow{A_iB_i} = \overrightarrow{OC} + \mathcal{R}\overrightarrow{CB_i} - \overrightarrow{OA_i} \tag{2}$$

For each distinct platform point $\overrightarrow{OB}_{i|_{R_f}}$ with $i = 1, ..., 6$, each kinematics chain can be expressed using the distance norm constraint, [42]:

$$L_i^2 = ||A_i B_i||^2 \tag{3}$$

C. The Inverse Kinematics Problem

Any rigid body can be positioned by 3 distinct points where variables have the same units and ranges and they include the rotation impact, [43]. The mobile platform distinct points are usually selected as the 3 joint centres B_1, B_2, B_3. The 9 variables are set as : $\overrightarrow{OB}_{i|_O} = [x_i, y_i, z_i]$ for $i = 1 \dots 3$. To simplify calculations, one reference frame b_1 is precisely located on B_1. The unit vectors u_1, u_2 and u_3 represent the new non-Cartesian frame axes, defined by:

$$u_1 = \frac{\overrightarrow{CB_1 CB}_{2|_O}}{||\overrightarrow{CB_1 CB}_{2|_O}||},\ u_2 = \frac{\overrightarrow{CB_1 CB}_{3|_O}}{||\overrightarrow{CB_1 CB}_{3|_O}||},\ u_3 = u_1 \wedge u_2 \tag{4}$$

Any platform point M can be expressed as $\overrightarrow{B_1 M} = a_M u_1 + b_M u_2 + c_M u_3$ where a_M, b_M, c_M are constants in terms of these 3 points. Hence, in the case of the Inverse kinematics problem (IKP), the constants are noted $a_{B_i}, b_{B_i}, c_{B_i}$, $i = i \dots 6$ and can explicitly be deduced from CB_C by solving the following linear system of equations :

$$\overrightarrow{B_1 B}_{i|_{R_{b_1}}} = a_{B_i} u_1 + b_{B_i} u_2 + c_{B_i} u_3 \ ,\ i = 1 \dots 6. \tag{5}$$

where a_{B_i}, b_{B_i} and c_{B_i} are parameters only depending of platform geometry. Using the relations in Equation(5), the distance constraint equations $l_i^2 = ||\overrightarrow{A_i B}_{iO}||^2$, $i = 1 \dots 6$ can be expressed. Therefore, for $i = 1 \dots 6$, the IKP is obtained by isolating the L_i actuator variables in the six following equations:

$$l_i^2 = (x_i - OA_{ix})^2 + (y_i - OA_{iy})^2 \ ,\ i = 1 \dots 3 \tag{6}$$

$$l_i^2 = ||\overrightarrow{B}_{k|_{b_1}} - \overrightarrow{OA}_{k|_O}||^2 \ ,\ i = 4 \dots 6 \tag{7}$$

D. The Forward Kinematics Problem

The IKP expression gives an algebraic system comprising the first six equations in terms of three point variables : $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$, given in Equation (7). This system contains trigonometric functions.

Being only applicable to optimization problems, the meta-heuristic techniques handle the objective function to be minimized. Therefore, we need to effectively convert the problem which solves a system of equations into an optimization problem. The inverse kinematic model is implemented from one single objective function, also called *fitness* function which is calculated on each FKP estimation representing 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. If we set $H_i = l_i^2$, from Equation (7), the fitness function is set to : $\sum_{i=1}^{6}(sqrt(H_i) - lg_i)^2$. This fitness function includes the combination of six individual objectives being the kinematics chain lengths. Preliminary tests led to several solutions which were NOT in correspondence with the exact proven ones. Hence, this function needed to be augmented by one constraint set : the platform fixed distances between the 3 selected joint points : B_1, B_2 and B_3 distinct points provide for 3 functions.

$$\begin{aligned} G_1 &= ||\overrightarrow{B_2 B}_{1C}||^2 - (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 \\ G_2 &= ||\overrightarrow{B_3 B}_{1C}||^2 - (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 \\ G_3 &= ||\overrightarrow{B_3 B}_{2C}||^2 - (x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2 \end{aligned} \tag{8}$$

Three distances can be calculated using 3 characteristic platform vector norms between the B_1, B_2 and B_3 distinct points.

$$d_1 = ||\overrightarrow{B_2 B}_{1C}||^2, d_2 = ||\overrightarrow{B_3 B}_{1C}||^2, d_3 = ||\overrightarrow{B_3 B}_{2C}||^2 \tag{9}$$

Hence, the fitness function becomes :

$$\sum_{I=1}^{3}(sort(H_I) - kg_I)^2 + \sum_{k=1}^{3}(G_I - d_I)^2 \tag{10}$$

This function includes the 9 single objectives obtained from the kinematical chain length norm and platform distance constraints.

TABLE I
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

IV. Results and Analysis

A. Configuration for the FKP of 6-6 Leg Parallel Manipulator

We shall examine one FKP example on a typical 6-6 parallel manipulator configuration, Table 1 shows the fixed base and mobile platform joint coordinates, $PA_{|_O}, CB_{|_C}$.

The prismatic actuator variables are set respectively to $\underline{L} = [1250, 1250, 1250, 1250, 1250, 1250]$. We have deliberately chosen one difficult case with 16 exact real solutions.

B. Experimental Set-up

In this section, the performance of simulated annealing, genetic algorithms and hybrid GA-SA is evaluated. The fitness function derived from the inverse kinematics of tripod 6-6 leg parallel manipulator given in Equation(10) is used. In all experiments, for GA and the hybrid GA-SA, roulette wheel selection is used in conjunction with the elitist strategy. A fixed population size of 40 individuals was used. Note that if the population size is P, then $2P$ selections are done in order to make P offsprings for the new population. In all SA set-ups, the cooling rate of 0.01 was used which indicated that the temperature was lowered by $T_0/100$ per chain, until it reached zero. The GA based algorithms aim in achieving the best error of the system, therefore, the inverse of the fitness is maximised. The following meta-heuristic strategies with the given configurations were used in order to find the best method for the problem:

1) **SA:** The simulated annealing algorithm used $T_0 = 1000$. The Markov chain length of 100 was used. These setting were determined by the optimal results given in trial experimental runs for the FKP problem.
2) **RCGA:** The real-coded genetic algorithm with Wright's heuristic crossover operator and Pivot mutation was used. The Pivot mutation operator proposed by the authors[13] have shown better performance when compared to non-uniform mutation for the forward kinematics problem. The crossover operator was employed at a rate of 0.9, and the mutation rate was 0.1.
3) **Hybrid GA-SA:** The genetic algorithm uses the Wright's heuristic crossover operator to build a single child from two parents. The simulated annealing and crossover parameter settings were the same as in **SA** and **RCGA**. The GA employs SA as a genetic operator with a probability of 0.9. The SA uses an initial temperature of 10 and N = 1000 iterations as shown in Step 4 of Algorithm(1). The value for N used in the SA process of the Hybrid GA-SA was determined in trial experimental runs with 100, 200, 500, and 1000 as the value for N. The best results was given when N was 1000, therefore this value was used in all experiments. Note that there is no mutation operator in the Hybrid GA-SA.

All experiments initialise solutions with real numbers in the range of $[-1000, 1000]$. The solutions contain 9 real variables which represent the positions x_i, y_i and z_i, where $i = 1, 2$ and 3. For each algorithm, a total of 100 experiments with different initial positions were done. The search terminated when the given search algorithms reached maximum training time given by iterations and generations, or when the fitness value gets lower than 0.01. An experimental run was considered successful when the algorithm terminated by obtaining an error (inverse-fitness) of 0.01 before the maximum training time is reached. Note that the given forward kinematics problem has 16 unique solutions.

The experiments were performed on an IBM compatible personal computer with 1.74 GHz dual core processors running Linux. The computation times are given in seconds. They were calculated using the standard C++ built in functions.

C. Results

Table II shows the mean and 95 percent confidence interval for 100 experimental runs for simulated annealing (SA), real-coded genetic algorithm (RCGA) and Hybrid GA-SA (Hybrid). The optimisation time is given by the number of iterations (No. Itera.) and CPU Time in seconds taken by the respective paradigm in converging to the required solution accuracy. The success rate shows how well the given paradigm can guarantee a solution when given any initial random solution within the search space.

TABLE II

META-HEURISTIC PARADIGMS FOR THE FKP

Algorithm	No. Itera.	Error	CPU Time(S)	Success
SA	7093± 1499	0.004 ±0.0006	3.19 ± 0.66	52
RCGA	5000±0	51.14±22.50	556.95±7.28	0
Hybrid	5±1	0.006±0.0003	56.39±4.66	100

D. Discussion

The genetic algorithm has failed to provide the solutions according to the required error, as is evident in the accuracy and success rates shown in Table(II) of the algorithms tested here. Simulated annealing has shown the best performance in terms of CPU computation time. However, the SA has a lower success rate when compared to the hybrid meta-heuristic approach. Note that all meta-heuristic approaches were able to report multiple unique solutions given multiple experimental runs with different initial search positions.

Therefore, the best paradigm in terms of solution accuracy and success rate for this particular problem is Hybrid GA-SA. Note that 14 out of 16 distinct solutions were reported from 100 experimental runs.

The unique solutions are obtained by running multiple experiments with different initial positions in the search space. In this way, the heuristic algorithm converges towards the solution nearest to the initial search position. For this reason, meta-heuristic techniques have their importance in solving the forward kinematics of parallel manipulators. Standard techniques such as gradient descent and Newton's Method fail to provide multiple solutions and are also prone to premature convergence when given an initial search position that is distant from the solution.

V. CONCLUSION

Although the simulated annealing algorithm is single solution based, it has performed the same task in a fraction of time taken by the genetic algorithm. The Hybrid SA-GA has given better solutions in terms of success rate.

This work has also shown that meta-heuristic algorithms, either single solution or population based, are able to provide different distinct solutions given multiple trial runs with different initial search positions. This is an advantage for optimisation problems where obtaining more than one distinct solution is important.

Meta-heuristic paradigms are easier to implement when compared to Newton's method and are independent of the problem domain. Therefore, the use of hybrid meta-heuristic paradigms is promising for optimisation problems in areas of robotics in general. In future work, it may be useful to use hybrid meta-heuristic paradigms in solving other problems such calibration and manipulator design.

References

[1] R. Boudreau and N. Turkkan, "Solving the forward kinematics of parallel manipulators with a genetic algorithmn," *Journal of Robotics Systems*, vol. 13, no. 2, pp. 111–125, 1995.

[2] T. J. Ypma, "Historical development of the newton-raphson method," *SIAM Rev.*, vol. 37, no. 4, pp. 531–551, 1995.

[3] D. S. Boudreau, R. and N. Turkkan, "Etude comparative de trois nouvelles approches pour la solution du problème géométrique direct des manipulateurs parallèles," *Mechanism and Machine Theory*, vol. 33, no. 5, pp. 463–477, 1998.

[4] M. B. A. Omran, G. El-Bayiumi and A. Kassem, "Genetic algorithm based optimal control for a 6-dof non redundant stewart manipulator," *International Journal of Mechanical Systems Science and Engineering*, vol. 2, no. 2, pp. 73–79, 2008.

[5] F. Ronga and T. Vust, "Stewart platforms without computer?" in *Proc. Int. Conf. on Real, Analytic and Algebraic Geometry*, 1992, pp. 197–212.

[6] D. Lazard, "On the representation of rigid-body motions and its application to generalized platform manipulators," *Computational Kinematics*, vol. 1, pp. 175–181, 1993.

[7] B. Mourrain, "The 40 generic positions of a parallel robot," in *Proc. ISSAC'93*, 1993, pp. 173–182.

[8] P. Dietmaier, "The stewart-gough platform of general geometry can have 40 real postures," *Adv. Robot Kinematics*, vol. 1, pp. 7–16, 1998.

[9] L. Rolland, "Certified solving and synthesis on modeling of the kinematics problems of gough-type parallel manipulators with an exact algebraic method," in *Parallel Manipulators, towards new Applications*. I-Tech Education and Publishing, 2008.

[10] H. Yu and W. Liang, "Neural network and genetic algorithm-based hybrid approach to expanded job-shop scheduling," *Computers and Industrial Engineering*, vol. 39, no. 4, pp. 337 – 356, 2001.

[11] G. Capi and K. Doya, "Evolution of recurrent neural controllers using an extended parallel genetic algorithm," *Robotics and Autonomous Systems*, vol. 52, no. 2-3, pp. 148 – 159, 2005.

[12] S. M.-S. B. M. B. Priauz, J. and E. Laporte, "Robust genetic algorithm for optimization problems in aerodynamic design," in *Genetic Algorithms in Engineering and Computer Science*. New York: Wiley, 1995, pp. 370–396.

[13] L. Rolland and R. Chandra, "Forward kinematics of the 6-6 general parallel manipulator using real coded genetic algorithms," in *Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2009)*, 2009, pp. 1637–1642.

[14] R. Chandra, M. Zhang, and L. Rolland, "The forward kinematics of 3rpr planer parallel manipulator using a hybrid meta-heuristic paradigm," in *Proc. of 8th IEEE International Symposium on Computational Intelligence for Robotics and Autonoumous Systems (CIRA 2009)*, 2009, p. In Press.

[15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[16] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, January 1985.

[17] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[18] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, "Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning," *Operations Research*, vol. 37, no. 6, pp. 865–892, 1989.

[19] D. F. Wong, H. W. Leong, and C. L. Liu. Kluwer Academic, Boston, 1988.

[20] R. Ghosh, M. Ghosh, J. Yearwood, and A. Bagirov, "Comparative analysis of genetic algorithm, simulated annealing and cutting angle method for artificial neural networks," 2005, pp. 62–70.

[21] T. B. Ludermir, A. Yamazaki, and C. Zanchettin, "An optimization methodology for neural network weights and architectures," *Neural Networks, IEEE Transactions on*, vol. 17, no. 6, pp. 1452–1459, 2006.

[22] J. H. Holland, *Adaptation in natural and artificial systems*. Cambridge, MA, USA: MIT Press, 1992.

[23] D. E. Goldberg, "Real-coded genetic algorithms, virtual alphabets, and blocking," *Complex Systems*, vol. 5, pp. 139–167, 1991.

[24] N. J. Radcliffe, "Equivalence class analysis of genetic algorithms," *Complex Systems*, vol. 5, pp. 183–205, 1991.

[25] J. D. Schaffer, R. A. Caruana, L. J. Eshelman, and R. Das, "A study of control parameters affecting online performance of genetic algorithms for function optimization," in *Proceedings of the third international conference on Genetic algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 51–60.

[26] J. Denzinger, M. Fuchs, and M. Fuchs, "High performance atp systems by combining several ai methods," in *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, 1997, pp. 102–107.

[27] E. K. Burke, G. Kendall, and E. Soubeiga, "A tabu-search hyperheuristic for timetabling and rostering," *Journal of Heuristics*, vol. 9, no. 6, pp. 451–470, 2003.

[28] M. Lozano and C. Garca-Martnez, "Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report," *Computers and Operations Research*, vol. In Press.

[29] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," Tech. Rep., 1989.

[30] Y. S. Ong, M. H. Lim, N. Zhu, and K. W. Wong, "Classification of adaptive memetic algorithms: A comparative study," *IEEE Transactions on Systems Man and Cybernetics – Part B*, vol. 36, no. 1, pp. 141–152, 2006.

[31] E.-G. Talbi and V. Bachelet, "Cosearch: A parallel cooperative metaheuristic," *Journal of Mathematical Modelling and Algorithms*, vol. 5, no. 1, pp. 5–22, 2006.

[32] F. Glover, J. P. Kelly, and M. Laguna, "Genetic algorithms and tabu search: hybrids for optimization," *Comput. Oper. Res.*, vol. 22, no. 1, pp. 111–134, 1995.

[33] H. Kim, K. Nara, and M. Gen, "A method for maintenance scheduling using ga combined with sa," *Comput. Ind. Eng.*, vol. 27, no. 1-4, pp. 477–480, 1994.

[34] R. Chelouah and P. Siarry, "Genetic and nelder-mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions," *European Journal of Operational Research*, vol. 148, no. 2, pp. 335 – 348, 2003, sport and Computers.

[35] R. Chandra and C. W. Omlin, "The comparison and combination of genetic and gradient descent learning in recurrent neural networks: An application to speech phoneme classification," in *Proc. of International Conference on Artificial Intelligence and Pattern Recognition*, 2007, pp. 286–293.

[36] S. W. Mahfoud and D. E. Goldberg, "Parallel recombinative simulated annealing: a genetic algorithm," *Parallel Comput.*, vol. 21, no. 1, pp. 1–28, 1995.

[37] A. H. Wright, "Genetic algorithms for real parameter optimization," in *Foundations of Genetic Algorithms*. Morgan Kaufmann, 1991, pp. 205–218.

[38] P. C. Pendharkar and J. A. Rodger, "An empirical study of impact of crossover operators on the performance of non-binary genetic algorithm based neural approaches for classification," *Comput. Oper. Res.*, vol. 31, no. 4, pp. 481–498, 2004.

[39] M. Raghavan and B. Roth, "Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators," *Trans. ASME*, vol. 117, pp. 71–79, 1995.

[40] L. Rolland, "Synthesis of the forward kinematics problem algebraic modeling for the general parallel manipulator: displacement-based equations," *Advanced Robotics*, vol. 21, no. 9, pp. 1071–1092, 2007.

[41] D. E. Parrish, R. and B. R., "An actuator extension transormation for a motion simulator and an inverse transformation applying newton-raphson's method," D-7067 NASA, Tech. Rep., 1972.

[42] J. P. Merlet, *Les Robots Parall'les*. Hermès, 1997.

[43] L. Rolland, "Certified solving of the forward kinematics problem with an exact method for the general parallel manipulator," *Advanced Robotics*, vol. 19, no. 9, pp. 995–1025, 2005.