GA Performance in Parametric Selection of Bridge Restoration Robot

A. McCrea - Senior Lecturer - School of Construction, South Bank University, 202, Wandsworth Road, London SW8 2JZ.
Researcher - Robotics Unit, City University, Northampton Square, London EC1, UK.
Tel: 0171-815-7210; Fax: 0171-815-7328;
E-mail: mccreaaa@sbu.ac.uk

Abstract

This paper is a follow-up to the last year's ISARC paper of mine and presents the process and the results of verifying the previously described concept (of using GA as optimisation method in parametric design of robots) using the customised Fortran GA driver. The adaptive search technique, known as the Genetic Algorithm (GA), is a powerful optimisation tool, and has been successfully applied to the preliminary and detailed engineering design. Here, we describe the results of using GA technique for optimising construction robot's kinematic parameters. The problem is investigated on an example leading to the development of an automated device to be used in restoration of steel bridges. This process is selected for robotisation due to its high health hazard when done conventionally, and because of its environmental impact. The discussion on effectiveness of the approach and GA performance conclude the paper.

1: Introduction

The initial step for any robot optimisation is to analyse the nature of the engineering design process and as a result, generate a geometrical description, known as configuration [Bramlette and Cusic, 1990]. Preliminary design results in the list of the selected parameters for optimisation and the optimal values for the selected parameters are to be determined.

One of the optimisation techniques is Genetic Algorithm (GA) which translates an engineering problem into a genetic one, where model characteristics are encoded in the form of genes and transmitted from one generation to the next. As the organisms evolve under the pressure of fitness proportionate reproduction, successive generations of the product or process under consideration are defined by different combinations of the design parameters, therefore producing results that tend to be increasingly fit for their purpose. This method ensures to a very high degree of probability, that the absolute optimum is found. Genetic Algorithm eliminates the inadequacies of purely numerical optimisation methods and proves to be a highly efficient one [Bullock et al, 1995].

2: Problem Statement

The paper addresses the parametric design problem using a design concept typical for 6 DOF robots to be employed for the task of paint removal off steel bridges. The robot's task is moving the blasting nozzle between points along given trajectories (generated as a result of the task analysis) at a constant distance from the surface. Therefore, the geometry of the part of the underside of a typical steel bridge (composite deck on primary and secondary Universal Beams) is modelled (using the computer graphic simulation package GRASP). This is done in order to geometrically position boundaries of the collision envelope (array of twenty corner locations encapsulating the envelope) and points along the paths required to be followed by the tool (four paths with three points each), with relation to the origin, later taken as the robot's base. Path points are allocated in such a way, as to enable to attend to the whole of the surface within the assumed, repetitive 'unit' workspace. Due to the symmetry of the geometry of this 'unit' workspace and the demanding computing capacity, the twelve points chosen, cater for half of the model's workspace. All points are located in relation to the origin, which is placed below the centre of the model workspace and is also taken as the centre of the robot's base.

The following kinematic design parameters are chosen for the optimisation: (i) the robot's main configuration (RRR with all three variables assumed as the angle movement range of three revolute joints and the spherical configuration - RRP, with two rotary motions and the third variable prismatic axial motion), namely the type and combination of the first 3 DOF, (ii) wrist configuration - last 3 DOF (RPR - Euler (spherical) wrist, performing in turn roll, pitch and roll motion and RPY performing first roll motion, followed by pitch and then yaw), (iii) optimal division of the unit length between two links in the RRR configuration, as it is an additional parameter for this configuration, compared to RRP, (iv) the joint working ranges, (v) joints' velocities and (vi) joints' accelerations. The criteria for optimisation are defined as: (i) collision avoidance, (ii) percentage of coverage, (iii) dexterity and (iv) productivity. From the logical point of view, it is possible to optimise all the parameters simultaneously as they are inter-dependent. However, due to their varied level of importance, it is preferable to group the criteria in two classes, as they govern different sets of parameters, and therefore divide the whole process into two stages.

2.1: First Stage Optimisation

In the first stage, the criteria of collision avoidance (the most important one) and percentage of coverage are addressed, and the optimisation of the robot aims at determining the best major configuration, the relationship between the link lengths in the RRR configuration and the optimal values of movement sectors for all joints.

The criterion of the percentage of coverage indicates the part of the work which will not be covered by the robot and therefore, will need further manual completion. The percentage of coverage is addressed through minimising the distance between each of the six points within the working environment (resulting from the task analysis) and the set of six locations of the tool, calculated with direct kinematic rules, using the parameters of the representation. The best configuration for the task is selected based on two separate computations carried out for both configurations and preference is established, based on the quantitative analysis of the objective functions representing the quality of results.

The first stage representation is separately developed for both configuration types and, as stated before, comprises twelve sets (four paths of three points each) of the first three joints' movement sectors and additionally for RRR configuration, a share in the unit length of the first link's length. The number of parameters varies for both configurations, because in RRP the third joint being prismatic, provides the reach, while in RRR configuration, all joints are angular and reach is additionally addressed, as a unity divided between two links.

Due to the different number of parameters, two separate computations have to be carried out for both configurations and a preference is established, based on the qualitative analysis of representations with the highest evaluation scores. This evaluation requires assessing the collision avoidance against given boundaries of the working environment (represented by externally input array of corner locations) and the percentage of coverage. To address the latter, the origin which is the robot's base, is placed centrally within the working space and 500 mm below the underside of the secondary steel beams, to imitate the robot's position on the mobile trolley. In order to clean within a typical bay, the nozzle has to run underside of the beams, along the sides of the beams and underside the deck (twelve points in total). As the choice of configurations is between the revolute RRR and the spherical one RRP, the validity of the outcome needs careful consideration. Ambiguity arises at this point, due to the fact that the number of 'independent' parameters is only four (joints' angular movement sectors and link length partition for RRR) and three (joints' movement sectors only), but the number of 'computed' parameters reflects the number of points to be reached (twelve), as well as the number of joints (three).

2.2: Second Stage Optimisation

The second stage involves optimising the parameters based on dexterity and productivity criteria. Here, the wrist type, the movement sectors of all six joints, and their optimal accelerations and velocities are optimised. In the second stage, the representation is using the already preferred main configuration, but is extended by the choice of the minor (wrist) configuration - responsible for the orientation of the tool (the last 3 DOF) and is followed by twelve sets (three points along four paths) of all six joints' movement ranges, velocities and accelerations. This allows us to evaluate the candidate solution with respect to dexterity and productivity. Compatibility of the dexterity between the candidate wrist embedded within the representation and the required one (resulting from the nature of the task and constraints of the environment), is achieved by comparing the relevant parts of the orientation matrices resulting also from direct kinematic calculations. The highest productivity is awarded to the representation requiring the shortest time for reaching and orienting the wrist in all twelve positions along the trajectories. This shortest time (smallest figure) is chosen out of the longest times, calculated separately for all six motors, as this simulates an assumed simultaneous movement, characterised by velocity and acceleration.

The criterion of productivity is addressed deficiently at this stage and offers only guidance to the overall task completion time, as the dynamics of the robot are not included in the analysis.

3: GA Approach

The Genetic Algorithm technique, as all evolutionary techniques requires a model of the system under design, so that relative fitness of designs based on different parameter combinations (represented in special, coded strings) can be determined. These strings also become candidate solutions to the problem. The representation of the design is a direct consequence of the choice of parameters for optimisation, while the criteria serve to evaluate the quality of the solution.

The basis of the success of the GA is the fact that the 'system' under design or optimisation is represented as a model consisting of all the parameters (currently optimised). Then, the pool of such representations is created (population) with the possible values of parameters randomly varying, although within the parameters' constraints. Each combination of the values of the parameters is submitted for evaluation, assessed in turn, and results in an allocated fitness value. Fitness or evaluation calculations are carried out through an objective function, embracing the criteria. The whole process combines the following: (i) rejection of parameter combinations, which produce low values and therefore, unfit designs, (ii) preferential reproduction of the more successful combinations and (iii) random generation of new values for further testing. All these operations enable the GA to sample varying areas of the design space whilst concentrating on highly fit regions.

A detailed description of the structure and use of the GA may be found in [Davies,1991].

For effectiveness, the parameters' values are often expressed in binary notation and are referred to as genes, which are combined into chromosomes and comprise a representation. If the representation is complex, it may even consist of a string of chromosomes. To commence a search of the design domain, an initial population of designs is produced by random generation of the population of chromosomes. Each member of the population is then evaluated by reference to the objective function. The next step involves the selection for fitness proportionate reproduction. Some designs are passed from one generation to another without modification, however among the candidates for reproduction, new combinations of parameters are created by use of crossover and mutation operators [Goldberg, 1989]. Although Genetic Algorithms using binary representation, single point crossover and binary mutation are robust, they are almost never the best algorithms to use blindly on any problem. Over the years, experiments on the effectiveness resulted in a variety of stochastic techniques for selection, the variations of the genetic operators trying different population sizes, development of the parameter sharing (niching) technique, etc. These are, however, subject of further implementation, as the one described here, is to prove suitability of the tool and the approach.

3.1: Evaluation and Fitness

Fitness of any potential solution is captured in an objective function which has the criteria, chosen in the conceptual stages of problem development, embedded in it. Dividing the optimisation process into two stages also helps the fitness function clarification. The first two criteria - collision avoidance and percentage of coverage are included into a single formula:

$$f1 = \frac{1}{\sum dist + I}$$

(1)

where, $\sum$**dist** is the sum of the distances between the arrays of points representing tool and path locations respectively. **I** is a penalty value added to each case of collision. Tool locations are obtained through direct kinematics calculations, using the parameters encoded in the representation (here, choice of the configuration type, movement sectors of the first three joints and the optimal division of the unit length between two links). Path points are externally allocated and reflect the robot's required tool's path (here - four paths of three points) with relation to the base. Each tool location in each representation is associated with consecutive points, respectively, along all the paths, mimicking the real movement of the tool and the sum of the distances is calculated each time. Collision is checked simultaneously. The sum of all these distances is added to the potential collision penalty and then, the reciprocal is taken, in order to favour the movement closest to the pre-allocated path, through the largest fitness function for the most favourable scenario.

The second two criteria - dexterity and productivity are assessed using separate computation, consisting of two independent formulas, which are then connected by being multiplied by 'weighting' factors (**A,B**), based on the relative importance of one criterion against the other.

$$f2 = A \times \frac{1}{(D-R)^2} + B \times \frac{1}{\sum tm}$$

(2)

D is the tensor of the rotational part of the total transformation matrix within the direct kinematics (for all six DOF), **R** is the tensor of the needed (result of the tool analysis, task requirements and workspace geometry) dexterity. The denominator in the second component - **tm**, stands for the sum of all the times needed to reach, in turn, every point along all paths from previous stage.

3.2: Choice of Genetic Operators

The members of the population evolve through generations, changing continuously using computing models based on operations that mimic the adaptive process of natural systems: selection for reproduction, crossover and mutation.

The competition for producing the next generation is achieved through binary tournament selection with a shuffling technique for choosing random pairs for mating. Pairs of individuals are chosen randomly from a population and the better out of the two is selected with fixed probability. In this implementation, each generation has the same size as the original one and if the best individual from the previous generation is not copied into the new one, a random member is replaced by it.

The choice of genetic operators depends on the encoding strategy. In the case of bit-string encoding, crossover and mutation are the most obvious ones. Traditional (single point) crossover is performed by choosing at random a single position in both parents and the parts after the crossover position are exchanged to form two new offsprings. Although, one-point crossover is inspired by biological processes, its algorithmic counterpart has drawbacks, as it cannot combine and protect certain combinations of features encoded in chromosomes. Therefore, different numbers of crossover points are experimented with, by GA practitioners. Hence, in order to link certain combinations of preferred features, a parameterised uniform crossover is introduced [Spears and De Jong, 1991]. Two offsprings are produced out of two parents with each bit position in both children being randomly decided, which parent it originates from. An exchange happens at each bit position when the probability test is passed. The success of the specific choice of the type of crossover depends, among other factors, on such ones as fitness function and type of encoding. Although, the software has an option for single point crossover, the uniform one is recommended and all the tests are carried out using the latter.

Although the crossover is considered the major instrument of variation and innovation in GA, mutation's importance as the tool against permanent fixation at any particular locus is widely recognised. In a simple GA, mutation is the occasional, with small probability, random alteration of the value of the string position and in binary coding it means changing a 1 to a 0 and vice versa [Goldberg, 1989]. When used with other operators it ensures that premature loss of vital information is avoided.

In this paper, a traditional jump mutation on a binary string is implemented and aided with creep mutation or real number creep [Davies,1991]. The idea behind the creep operator is that a chromosome which is reproducing is already in a fairly good position in relation to other members of the population. What is needed, is just a small browse around the current position to see if a movement nearer the optimum can be detected. The creep mutation moves along the chromosome, creeping up or down each parameter by an increment, by which the parameter array is increased. This is achieved by converting the binary encoding into a real number, creeping and converting back.

4: Fortran GA Driver Application

The paper uses an adapted version of a FORTRAN genetic algorithm (GA) driver [Carroll, 1996].

The output of the first-stage run shows the average value of each parameter in each generation, the average fitness value, the best fitness of the generation, number of crossovers, jump and creep mutations and number of elitist reproductions. The average fitness per generation is plotted against the generation number and the optimisation for the first stage is initially carried out for 1000 and 4000 generations. It demonstrates standard behaviour, schematically shown in Fig.1.

With the limits for some of the parameters (link length and major joints' ranges) refined from the first stage, the second stage representation is run also for 4000 generations and the relevant figures represent the variations of the average fitness through the generations.

The algorithm also indicates when the convergence is at the level of 20% and when there is no change in the best member over 50 consecutive iterations. These figures are purely empirical and act as the indication of potential optimum, which in turn needs closer investigation of its parameters.

4.1: Genetic Parameters

Setting the values for genetic parameters, such as population size, crossover and mutation probabilities has to be made (similarly to the choice of genetic operators), based on literature review and then trial and error, as these parameters interact with each other non-linearly and therefore, cannot be optimised one at a time.

Values of the critical variables are set up as follows:

(i) population size - 1000, as for large problems like this, a hundred individuals (as recommended by [Goldberg, et all 1992]) is not enough;

(ii) probability of jump and creep mutation are: 0.01 and 0.06 respectively, as this relationship generates (using basic probabilistic arguments) approximately the same number of creep and jump mutations per generation; and

(iii) probability of uniform crossover - is assumed as 0.9. This is quite a high rate, but it shows in practice not only a rapid improvement in the fitness value, but also a steady climb-up afterwards.

5: Results of GA Performance and Conclusions

There are several procedures (e.g. enumeration, machine learning, or artificial intelligence), which would lead to global optimum, but they require excessive computing capacity and time to set them up. Effectiveness of genetic algorithms can be assessed, however, at early and intermediate stages, and in many various ways. As suggested by [Holland,1975], when searching large finite spaces, convergence is not the most useful performance measure, as there is always a danger that the optimum is not the global but the local one. To avoid the search algorithm being entrapped in a local optimum, various methods are available, such as (i) improvements to the searching mechanisms, (ii) observing the speed with which the optimum is arrived at, (iii) analysing the efficiency of the fitness function with which it approaches the optimum or (iv) the analysis of the quality of the optimum solution at the intermediate stages.

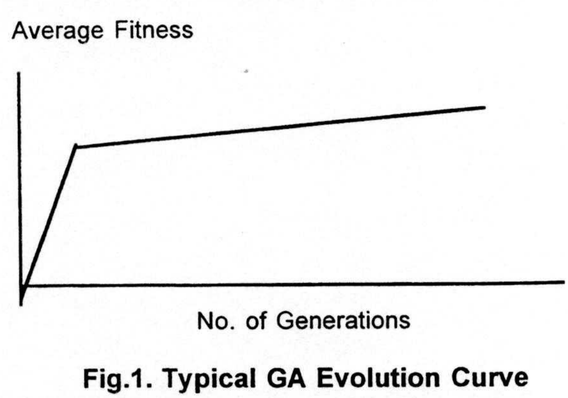

Fig.1. Typical GA Evolution Curve

As the problem under investigation is a complex one and involves searching large spaces, the approach of monitoring the performance throughout and analysing the current optima, is adopted. The typical form of the evolution curve is shown in Fig.1 from which it is evident that the major improvements tend to occur during the early stages of search. Progression beyond the turning point on the curve, often requires the introduction of increasingly sophisticated control parameters, for even small gains. Finalising a large optimisation problem requires significant computing capacity, while just monitoring the development of the best individual through generations can supply the indicative information or even the required outcome at earlier stages.

Monitoring the output through the distribution of the average fitness only, gives information about the speed and convergence profile towards the optimal solution. Assessment of the best individuals proves to be of the greatest value. It is vital to notice that even tiny improvement in the representation's maximum fitness can bring significant changes in the values of the parameters. Also introducing standard deviation calculations gives additional information about the quality of the solution. When the improvement in fitness is accompanied by an increase in standard deviation, calculated for the distances between tool positions and a pre-determined path, a further study may be needed to confirm the quality of the best representation. However, first of all, careful analysis of the well performing individuals in the population in both computations (for RRR and RRP configuration) has to determine the more suitable configuration for the task. The calculations for both configurations relate numerically, as the total link length in RRR and maximum value of the reach parameter in RRP are both 1.0, revolute joints have the same movement sector ranges and the workspace and the position of the robot are also identical. Therefore, it is rational to compare both performances also numerically. Both programs are run for 500 generations, as the major growth is inclined to occur during early stages. Initially, the average fitness (and convergence performance) does not dramatically improve in both configurations, however, this is to be expected, as the number of parameters is quite significant. Additionally, only the preferential performance profile for two configurations is anticipated, so arriving at the optimum is not the primary aim at this stage. Further analysis indicates, through a numerical comparison of the best fitness and convergence level in both configurations, that the RRP is better performing and therefore, more suitable for the task. Hence, the RRP is the one which is admitted to the second stage which is run for another 500 generations. The number of generations is purely empirical, it is noticed that the fitness is continuously improving and with such a large problem rational decision about how much computation effort to expend in trying to improve the design and performance of a particular system, can be made only on the economic and common sense basis. The last representation giving the maximum fitness is examined and the boundary values of all three parameters are then outlined to show the movement sectors for all three joints (to determine the percentage of coverage). This information may additionally be used to determine the choice of motors and for construction purposes.

The second stage representation uses the reduced boundaries of the joints' movement sectors for the first 3 DOF (taken from the first stage) and is run also, for 500 generations, to show the behaviour of the average fitness throughout the generations. It becomes evident that, as the total fitness function consists of two independent criteria, it would be beneficial to plot separately the average fitness due to dexterity and productivity. Similarly to the first stage, the performance of the best individual at the last computation is closely investigated. Using the previously mentioned formula for the population size [Goldberg, et all 1992], the most efficient size for the second stage's problem is of approx. 4000 members in the population. Due to the scale of this computation, a reduced size of 1000 is used, hence a poorer quality solution is obtained at the 500th generation. It is, however, possible to detect significance in the findings based on the results achieved so far.

Analysis of the best individual allows us to draw several conclusions. The superior configuration for the task is clearly identified, as the spherical one and initial verification using computer simulation and inverse kinematics indicates, that it is clearly the better choice of the two. Then, the ranges of the movement sectors of the major configuration's joints are identified. This information not only helps to calculate the percentage of manual involvement but can also assist in the kinematic design and choice of actuators. The second stage is using previously refined ranges of the parameters, for the second set of criteria. Additional parameters involve the preferred choice of the wrist configuration, which is the Euler wrist (RPR), the movement sectors of the joints within wrist which, in turn, can help the choice of the actuators and the geometry of the end effector. The most economical values for the velocities and accelerations reinforce the preference for the joints' actuators and help to calculate cost of running the robot.

The scale of the computations, however, restricts the simplicity and efficiency in obtaining final results at this stage. Therefore, further study is carried out into the micro-GA population and niching which significantly reduces the population size together with the computing requirements and therefore allows further exploration of the peripheral regions of the search space.

6: References

Bramlette, M.F. and Cusic, R.A. (1990). A comparative evaluation of search methods applied to parametric design of aircraft, Proceedings of The Third International Conference on Genetic Algorithms.

Bullock, G.N., Denham, M.J., Parmee I.C. and Wade J.G. (1995). Developments in the Use of the Genetic Algorithm in Engineering Design, Design Studies, 16(4): 507-524.

Carroll, D.L. (1996). The General Genetic Algorithm (GA) Driver, E-mail: carroll@uxh.cso.uiuc.edu

Davis, L. (ed) (1991). Handbook of Genetic Algorithms. New York: Van Nostrand Reinhold.

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimisation and Machine Learning. Reading, MA: Addison-Wesley.

Goldberg, D.E. (1990). A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-

Oriented Simulated Annealing, Complex Systems, Vol.4, Complex Systems Publications, Inc., pp.445-460.

Goldberg, D.E. and Deb, K. (1991). A Comparative Analysis of Selection Schemes Used in Genetic Algorithms, Foundation of Genetic Algorithms; Rawlins G.J.E., ed., Morgan Kaufmann Publishers, San Mateo, CA, USA: 69-93.

Goldberg, D.E., Deb, K. and Clark, J.H. (1992). Genetic Algorithms, Noise and Sizing of Populations. Complex Systems, Complex Systems Pub. Inc., 6:333-362.

Holland, J.H., (1975) Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press.

Klein, Ch.A. and Blaho, B.E. (1987). Dexterity Measures for the Design and Control of Kinematically Redundant Manipulators. The International Journal of Robotics Research, 6(2): 72-83.

Spears, W.M. and De Jong, K.A. (1991). On the Virtues of Parameterized Uniform Crossover, Proceedings of the Fourth International Conference on Genetic Algorithms, Belew, R.K., and Booker, L.B., eds, Morgan Kaufmann Publishers, San Mateo, CA, USA: 230-236.