Abstract: The paper proposes the design of the intelligent robot system for resistance welding. The robot system incorporates the optimization module based on the swarm intelligence approach. The module is intended for optimization of robot path during welding.

Key words: Robotics, Robot cell, Optimization algorithm, Swarm intelligence, Ant colony optimization.

1. INTRODUCTION

Robotics is gaining increasing importance in the modern world, since the demand for robots is growing due to opening of new, quickly growing areas of their use. At the beginning, as much as three thirds of robots were used, particularly for welding in the automobile industry, while now they are used also in other industrial branches, such as electronics, production and processing of food and drinks, pharmaceutics, production of household appliances etc. They are used wherever high quality of products is required, where the working operations are monotonous and harmful to health.

The basic reasons for automation and robotization are cost reduction, relief of people and assurance of adequate capacity and quality of production. The automation and robotization have a considerable influence on shortening of the manufacturing time, increasing of the capacity and reducing of production costs, which, however, is often hard to calculate accurately in advance and sometimes hard to justify.

The companies view the automation and robotization mainly from the stand point of savings and costs, but increasingly also as a chance to remain competitive in a certain industrial branch.

The paper proposes the design of the intelligent robot system by the swarm optimization of resistance welding application of the older ACMA XR701 robot made by Renault.

Swarm optimization is a swarm intelligence based algorithm for finding a solution to an optimization problem in a search space or model, and predict social behavior in the presence of objectives.

Swarm optimization is a stochastic, population-based evolutionary computer algorithm for problem solving. It is a kind of swarm intelligence that is based on social-psychological principles and provides insights into social behavior, as well as contributes to engineering applications. The swarm optimization algorithm was first described in 1995 by James Kennedy and Russell C. Eberhart. The techniques have evolved greatly since then, and the original version of the algorithm is barely recognizable in the current ones.

Intelligent optimization methods are finding their way into more and more diverse areas of human activity. The reason lies in the increase of the capabilities of computer systems and, consequently, in the increase of opportunities for the development and use of artificial intelligence systems (see, e.g., [1-2]).

Taking into account the problem encountered in designing of the welding application, the paper proposes the optimization algorithm with which the optimum time of transition of the electrode holder between welding spots will be reached.

When comparing the optimization algorithms, the analysis of optimization algorithms and methods was made and on the basis of the findings it was decided to use the optimization algorithm based on swarm intelligence with particle colony algorithm.

2. OPTIMIZATION WITH SWARM INTELLIGENCE (Particle Swarm Optimization)

This section is aimed at presenting the optimization with swarm intelligence and at describing the functioning of the optimization process.

The particle swarm optimization algorithm (PSOA) is the optimization algorithm based on the population stochastic optimization technique. The PSOA was developed by J. Kennedy and R.C. Eberhart in 1995, when they studied the social behaviour of swarms of birds and fish.

The PSOA is very similar to the genetic algorithm. In case of both methods (PSOA and GA) the optimization starts with randomly selected members of population searching for optimums with constant improvement of the subjects of the population in the course of improvements from generation to generation. Unlike the GA, the PSOA does not have evolutionary operators, such as crossover and mutation, which considerably simplifies the PSOA in comparison with the GA. In case of the PSOA the population members are called particles (to which the method owes its name) moving in the space of solutions in such a way that they follow the best particle. Like the GA also the PSOA is used for similar cases: optimization of functions, learning of neural networks, for setting of parameters of controllers by fuzzy logic etc., i.e., wherever the GA has already been well established.
2.1 How PSOA acts

The PSOA simulates the behaviour of swarms of birds or also insects. For example, let us imagine a swarm of birds randomly examining the space when looking for food. In the examined space there is only one source of food. None of the birds know where the food is, but each bird knows how far away from food it is. So, what is the best strategy of each individual bird in the swarm? The most effective strategy is to follow the bird nearest to the food.

The PSOA assumed the searching technique for solving optimization problems. In the PSOA each possible solution (bird, ant, bee) is called particle. All particles have the fitness value (distance of each particle from the food source) calculated by the fitness function for the purpose of optimization. Each particle also has its speed controlling "flying" of particles towards the optimum. The particles fly through the space of solutions so that they follow the currently most optimal particle [3-6].

The PSOA is initialized with the swarm of randomly placed particles into the space of solutions, afterwards searching for optimum from iteration to iteration (generation). In each iteration, each position of the particle is calculated again on the basis of three "best" positions of particles stored from iteration to iteration. The first best position of the particle is the position reached out of all hitherto iterations for the individual particle - it is called the local best (lbest). The next best position of the particle is the position reached out of all hitherto iterations among all particles - it is called the global best (gbest). The last best position of the particle is the position reached out of all hitherto iterations among the neighbouring particles - it is called the neighborhood best (nbest).

After searching for the said three best values (lbest, gbest and nbest), the particle speed is calculated for the particle: 

\[ v_{ij}(k) = c_0 * v_{ij}(k-1) + c_1 * \text{rand}(0,1) * (g_{bestj}(k-1) - x_{ij}(k-1)) + c_2 * \text{rand}(0,1) * (l_{bestj}(k-1) - x_{ij}(k-1)) + c_3 * \text{rand}(0,1) * (n_{bestj}(k-1) - x_{ij}(k-1)) \]

(1)

and

\[ x_{ij}(k) = x_{ij}(k-1) + v_{ij}(k). \]

(2)

The function \( \text{rand}(0,1) \) calculates the random value between 0 and 1 including 0 and 1. The constant \( c_0 \) represents the value of speed of particle from previous iteration and represents the particle movement inertia. The usual value of \( c_0 \) is between 0 and 1 (best values are slightly below 1). The constants \( c_1, c_2 \) and \( c_3 \) are the learning constants and, usually, have the value of approximately 2. The calculation of the fitness function represents the conversion of the j-dimensional information about the particle position in the space of solutions into a one-dimensional scalar value representing the quality of the particle position and/or its distance from the optimum.

The simple pseudo code for building of the computer programme of PSOA is as follows:

FOR each particle

Initialize particle

END

DO

FOR each particle

Calculate fitness value

IF fitness value is better than \( l_{best} \)

set current value of particle as new \( l_{best} \)

END

END

Select particle with best fitness value of all particles in hitherto iterations as \( g_{best} \).

FOR each particle

Calculate particle speed by equation (1).

Calculate new particle position by equation (2).

END

WHILE maximum of iterations or minimum criterion has not been reached.

3. TRAVELLING SALESMAN PROBLEM

Optimum movement between welding spots represents the travelling salesman problem. Therefore, this section of the paper presents the theoretical and applicative approach to solving the optimization travelling salesman problem with the algorithm of ant colony optimization for the case of resistance spot welding application.

3.1 Solving minimum path in graph

The theoretical approach requires the specification of the facts important in searching for minimum paths in the graph.

Similarly as in case of natural ants, the task of the artificial ants is to find the minimum path between points in the graph. For example, the set \( G=(N,A) \) represents the linked up graph, where the number \( N \) is the number of all points of the graph and the number \( A \) is the number of all paths. The number of points is \( |N| = n \). Solution of the task (Fig.1) is the path in the graph connecting the starting point \( f \) with the desired point \( d(\text{target}) \), the length being determined with the number of transitions on the path [7-9].

Variable \( \tau_{ij} \), representing the artificial trace of the
feromone, is connected with each transition \((i,j)\) in graph \(G\). Thus, the ants leave the trace of feromone. In each point of the graph the stochastic (random) decision as to which point is the next occurs. The \(k\)-th ant placed in point \(i\) uses the feromone trace \(\tau_{ij}\) for the calculation of probability to which point \(j \in N_i\) it is necessary to go. \(N_i\) is the group of neighboring points \(i\). The probability of selection of point \(j\) is given with the term (3):

\[
 p^k_{ij} = \begin{cases} \frac{\tau_{ij}}{\sum_{j \in N_i} \tau_{ij}}, & j \in N_i \\ 0, & j \notin N_i \end{cases} \tag{3}
\]

As long as there is a solution, the ant leaves the feromone trace on the existing path \(\tau_{ij}(t) \leftarrow \tau_{ij}(t-1) + \Delta \tau\). However, the solution so defined can lead to convergence towards optimum solution, if the mechanism of feromone evaporation is introduced. In each interaction the quantity of feromone is exponentially reduced according to the following term (4):

\[
 \tau \leftarrow (1-\rho)\tau, \quad \rho \in [0,1] \tag{4}
\]

The behaviour of the algorithm depends on the number of neighbors of each point. If the points have more than two neighbors, the algorithm loses its stability.

Consequently, also the choice of parameters becomes difficult. The algorithm can be improved by changing the properties of the colony and by changing the individual ants in the colony.

For testing of the ant colony algorithm the algorithm, by which the above theoretical findings were verified also in the simulation, has been created in the programme environment MS Visual Studio.Net. The graphic environment has been created, with which solving of the travelling salesman problem between randomly placed spots was simulated. By changing the ant colony algorithm parameters various mutual dependences of the ant colony parameters were simulated [10].

4. SIMULATION EXAMPLES OF THE WELDING APPLICATION OPTIMIZATION

The following are different examples of use of the optimum programme with ant colony. Particularly interesting is the incorrectly arranged configuration of placing of spots (random placing). It can be noticed on the final result that it is not deterministically determined and that the case can arise, when two output solution do not give identical solutions with the same number of spots and with the same parameters.

4.1 Incorrectly/randomly placed spots

In case of incorrect placing, the target function has a great number of local optimums and it may happen that the algorithm gets “caught” into the local optimum. As even the optimization with the ant colony algorithm does not represent an exact solution of optimization, it may happen that the final result is not the optimum solution. This can be seen in the examples in Fig. 3 and 4.

Fig. 3. First solution of travelling salesman problem

The first solution of the travelling salesman problem is the solution with random placing of spots in iteration 12 and with adjusted parameters \((\alpha = 1, \beta = 0, \text{Rho} = 1)\), the final solution being the length of 303333 units.

Fig. 4. Second solution of travelling salesman problem

The second solution of the travelling salesman problem is the solution with random placing of spots in
iteration 11 and with adjusted parameters ($\alpha=1$, $\beta=0.5$, $\rho=1$), the final solution being the length of 69213 units, which is a better solution than in the first case.

The layout of spots has a very important influence on the behaviour of the optimization algorithm of the ant colony. If the spots are correctly (circularly) arranged, the algorithm converges very quickly, irrespective of the spot number. In this case it is not necessary and/or not reasonable to use large colonies of ants (great number of ants).

Also the analysis of mutual influences of the parameters alpha, beta and Rho is interesting, since each of them has an important influence on the problem solution. The analysis of said parameters, mutual influences and the influence of the ant number on the final solution will be extensively included in further research work [11].

5. CONSLUSION

The paper has presented an example of building of the intelligent robot system by the swarm intelligence. The central part of the research has been devoted to the area of use of swarm intelligence for optimization of the robot welding system. The primary design concepts of the swarm intelligence are adopted from the world of animals, whose life cycle associates them into colonies, by copying the laws from the natural life of swarms into the computer artificial life of the swarm, called particle swarm theory, and thus a computer algorithm for solving optimization problems in robot systems is obtained.

Through building of the applicative programme environment, the findings of some well-known authors (Dorigo, M., Gambardella, L.M.), stating that the spot arrangement importantly influences the behaviour of the optimization algorithm of ant colony have been confirmed. If the spots are correctly (circularly) arranged, the algorithm converges very quickly, irrespective of the spot number. In this case it is unnecessary and/or inappropriate to use large ant colonies (great number of ants). When the spot arrangement is incorrect/random, it is already more contestable whether the algorithm is optimum. Also the final solution is not unambiguously determined.

The findings reached by simulations have been successfully incorporated into the robot welding application and the minimum time of transition of robot electrode holder between the individual welding spots has been assured.

6. REFERENCES


Authors: Assoc. Prof. Dr. Miran Brezocnik, Simon Brezovnik B.Sc, Prof. Dr. Joze Balic, University of Maribor, Faculty of Mechanical Engineering, Smetanova 17, 2000 Maribor, Slovenia, Prof. Dr. Bogdan Sovilj, University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovica 6, 21000 Novi Sad, Serbia, E-mail: mbrrezocnik@uni-mb.si simon.brezovnik@uni-mb.si joze.balic@uni-mb.si bsovij@guns.ms.ac.yu