Article ID: 1000 - 8152(2003)05 - 0700 - 07

Two new algorithms for minimum time motion path planning of robot manipulators and their implementations on PVR-based control platform

LUO Xiong, FAN Xiao-ping
(College of Information Science and Engineering, Central South University, Hunan Changsha 410083, China)

Abstract: To deal with the minimum time motion path planning (MTMPP) of robot manipulators to execute point-to-point work task, a novel hybrid evolutionary computation simulated annealing algorithm EC* SA and an algorithm EC* SA_DP combining EC* SA and some special optimization techniques were presented. The advantages of novel optimal algorithm were demonstrated through a comparison with existing preferable elastic net method (ENM) and some other optimal algorithms such as simple genetic algorithm (SGA) and simulated annealing (SA) algorithm. In addition, virtual simulation experiments were carried out on the control platform based on projective virtual reality (PVR) technology. The virtual experimental results show that the precision of projective operability is greatly enhanced in the virtual environment.

Key words: minimum time motion path planning; evolution computation; simulated annealing; elastic net method; simple genetic algorithm; projective virtual reality

CLC number: TP241.2　　**Document code**: A

两种求解机械手最短时间动作路径规划的新算法及其在基于 PVR 技术的控制平台上的实现

罗　熊，樊晓平
(中南大学 信息科学与工程学院，湖南 长沙 410083)

摘要：针对机械手在执行点到点的工作任务中所遇到的两类最短时间动作路径规划(MTMPP)问题，分别提出了新的混合型进化计算模拟退火(EC* SA)算法以及将 EC* SA 算法与一些优化技术结合使用的 EC* SA_DP 算法．通过与目前较好的求解算法(如弹性网络方法 ENM)以及其它一些近似优化算法(如遗传算法和模拟退火算法等)所进行的数值仿真比较，验证了 EC* SA 算法在处理复杂工作任务时的高效性．这些算法在基于投射式虚拟现实(PVR)技术的控制平台上进行了虚拟仿真实现，仿真结果表明可有效地提高虚拟环境中的投射式操作精度．

关键词：最短时间动作路径规划；进化计算；模拟退火；弹性网络方法；简单遗传算法；投射式虚拟现实

1 Introduction

The minimum time motion path planning (MTMPP) of robot manipulators is always the hot spot in the research field of robots in the foregone forty years.

There are some sophisticated and high-powered optimal control algorithms concerning MTMPP of robot manipulators operating with relatively simple motions[1~3]. These algorithms can find the minimum time motion along a specified path for a manipulator. To the end, many researchers pay much attention to the manipulator's path planning task called point-to-point (PTP). It means that the manipulator has to move between work points and must reach and stop at each work point. In fact, the optimization of PTP task can be treated as the well-known traveling salesman problem (TSP) approximately. Some representative research achievements can be found at [4, 5].

Unfortunately, however, there are some common drawbacks and deficiencies in these existing literatures, which can be generalized as follows:

Received date: 2002 - 12 - 16; Revised date: 2003 - 06 - 10.
Foundation item: supported by the National Natural Science Foundation of China (69975003); the Foundation for University Key Teachers by the Ministry of Education of China ([2002]65).

1) The number of work points was considered to be relatively small, and some precise solution algorithms for TSP were used, which restricted the application scopes of the algorithms.

2) The mathematic model used in this problem was too simple and the optimization problem was simplified far more excessively, which could not embody the essential characters of the initial problem completely.

In order to get over the lack of earlier approaches, the optimal problem is considered in this paper thoroughly and novel hybrid evolutionary computation simulated annealing algorithms EC* SA and EC* SA_DP are presented.

Initiated by Freund and Rossmann in 1997, projective virtual reality (PVR) technology is a new approach used to control and supervise robots and other automation systems[6]. PVR can get over many restrictions existing in most other teleoperation approaches presented so far. In order to provide the supervision capability and further simplify users' jobs to command different tasks, many images and metaphors in the virtual world had been developed to support the intuitive operability and control[6]. In this paper, on the PVR-based control platform, the graphic simulation software is developed using the proposed new algorithms, and a useful metaphor is gotten to strengthen and improve the performance of PVR systems.

2 Problem statement

The PTP-oriented MTMPP can be regarded as the TSP approximately. The presuppositions for the application of TSP in MTMPP of PTP task can be generalized as follows.

1) There are N work points in the workspace that must be visited by manipulators one by one when it executes the given work task.

2) There was a presupposition appeared in [5]. Namely, the visiting order of N work points is unrestricted, i.e., the precedence relationship to visit two arbitrary points does not affect the accomplishing of the work task. However, it restricts the application area of former algorithms.

Here the proposed algorithms will be used not only in unrestricted but also in restricted visiting. The details should be presented in the next section.

3) The discussed robot manipulator is non-redundant. Its degree-of-freedom (DOF) is n. The number of dimensions of workspace is unrestricted and denoted as T (T usually be 3).

The goal of the algorithms presented here is to find an optimal cycle path for manipulators to visit N work points.

3 Algorithms for PTP-oriented MTMPP of robot manipulators

3.1 With unique configuration in each work point

In this case, N work points $\Theta = (C_{\Theta,1}, C_{\Theta,2}, \cdots, C_{\Theta,N})$ are given, where, $C_{\Theta,i}$ has only an unique configuration $(1 \leqslant i \leqslant N)$.

Based on the algorithm by Luo and Fan[7], a novel hybrid evolutionary computation simulated annealing algorithm EC* SA for this class of MTMPP is presented as follows.

Algorithm EC* SA

1) $k \leftarrow 0$; //Initialize counter

2) Initialize (Temperature (k));
   //Initialize annealing temperature

3) //Generate the initial population. Here, an individual is a motion cycle path of manipulator
   //which is just one of the arrangements for work
   //points $(C_{\Theta,1}, C_{\Theta,2}, C_{\Theta,3}, \cdots, C_{\Theta,N})$.
   Initialize (Population (k));
   $// = (P_k(1), P_k(2), \cdots, P_k(N))$

4) EvaluateObjectFunction (Population (k));
   $// = (f(P_k(1)), f(P_k(2)), \cdots, f(P_k(N)))$

5) EvaluateFitness (Population (k));
   $// = (F(P_k(1)), F(P_k(2)), \cdots, F(P_k(N)))$

6) TerminationCriterion = FALSE;

7) While (! TerminationCriterion)
   {

8) $k \leftarrow k + 1$;

9) Population $(k-1)^{(1)}$ ←Select (Population ($k-1$)); //Selection Operator

10) Population $(k-1)^{(2)}$ ←Crossover (Population $(k-1)^{(1)}$); //Crossover Operator

11) Population $(k-1)^{(3)}$ ←Mutation (Population $(k-1)^{(2)}$); //Mutation Operator

12) Population $(k-1)^{(4)}$ ←Inversion (Population $(k-1)^{(3)}$); //Inversion Operator

13) //Generate new individual from the old one

// according to certain strategies.
// Those individuals will be accepted or refused
// according to Metropolis Criterion.

14) for ($i = 1; i <= N; i ++$)
{
15) Generate $P_{k-1}(i)'$ from $P_{k-1}(i)^{(4)}$;
16) // Strategy I: For certain consecutive sections of
// the current motion path, those draw-out points
// are rearranged with the
// contrary order corresponding to the quondam
// order and those points are used to replace the
// former ones.
// Strategy II: Delete certain consecutive sections
// and use the other generated randomly consecu-
// tive sections of the
// current motion path to replace the deleted sec-
// tions.
// The above two strategies can be chosen rando-
// mly in each generation process.
17) if ($(f(P_{k-1}(i)') \leqslant f(P_{k-1}(i)^{(4)}))$)
18) Population $(k-1)^{(4)} \leftarrow$ Population $(k-1)^{(4)} \cup \{P_{k-1}(i)'\}$;
19) else if (exp $(-(f(P_{k-1}(i)') - f(P_{k-1}(i)^{(4)}))/$ Temperature $(k-1)) >$ random $[0,1)$)
20) Population $(k-1)^{(4)} \leftarrow$ Population $(k-1)^{(4)} \cup \{P_{k-1}(i)'\}$;
}
21) $N' \leftarrow |$Population $(k-1)^{(4)}|$;
22) for ($i = 1; i <= N'; i ++$)
// Do the stretch for the fitness of individual. (*)
23) $F(P_{k-1}(i)^{(4)}) \leftarrow \exp(F(P_{k-1}(i)^{(4)})/$ Temperature $(k-1))/(\sum_{j=1}^{N'} \exp(F(P_{k-1}(i)^{(4)})/$ Temperature $(k-1)))$
24) // Generate weight values for individuals in Popu
// lation $(k-1)^{(4)}$ according to the comparison
// in it. (* *)
25) for ($i = 1; i <= N'; i ++$)
{
26) Select q random individuals from Population $(k-1)^{(4)}$; // = $(P_{k-1}(1)^{\#}, P_{k-1}(2)^{\#}, \cdots,$
// $P_{k-1}(q)^{\#})$, where $1 \leqslant q \leqslant N'$.
27) number $(i) \leftarrow \sum_{j=1}^{q} \lfloor F(P_{k-1}(i)^{(4)}/\max\{F(P_{k-1}(i)^{(4)}), F(P_{k-1}(j)^{\#})\} \rfloor$; // $\lfloor \cdot \rfloor$ is the symbol of rounding.
// number (i) denotes the number of the individ-
// ual $P_{k-1}(j)^{\#}$ whose fitness is smaller than
// $F(P_{k-1}(i)^{(4)})$.
28) Weight $(P_{k-1}(i)^{(4)}) \leftarrow$ number(i);
}
29) Population $(k-1)^{(5)} \leftarrow$ Sort descending for N' individuals of Population $(k-1)^{(4)}$ according to Weight $(P_{k-1}(i)^{(4)})$;
30) Population $(k) \leftarrow$ Select the first N individuals of Population $(k-1)^{(5)}$;
31) Temperature $(k) \leftarrow$ Update (Temperature $(k-1)$);
32) EvaluateObjectFunction (Population (k));
33) EvaluateFitness (Population (k));
}
34) Export the best individual (i.e. the sequence of N work points) as the minimum time motion cycle path for robot manipulators.

In the above algorithm, four operators are used: proportional model select operator; edge recombination crossover (ER) operator; repeated exchange mutation (EM) operator and inversion mutation (IVM) operator. In addition, several special operators or transforms are designed.

Remark 1 In operator (*) of the above algorithm, the stretching or scaling for fitness is introduced[8]. When the annealing temperature descends, the effect of scaling or stretching is strengthened, and the difference among the individuals with close fitness is magnified, which makes the advantage of excellent individuals more obvious.

Remark 2 In operator (* *) of the above algorithm, a special selection operator is introduced. This selection operator embodies the competition of individuals, which can always let several best individuals get inherited in the next population.

Remark 3 A convergence conclusion can be made by the properties of Markov chains. That is, the probability of EC* SA's converging to a set of optimal population consisting of global optimal solutions will finally approach 1. This conclusion can be proved similarly as in [9]. As for the analysis of convergence rate, it is easy to draw a conclusion based on the figure descriptions for the convergence process in Section 4. For the sake of paper length, it should be not discussed in detail.

Remark 4 An initial time cost matrix is constructed. It should note that algorithm EC* SA can be used only in the "complete graph" which means the manipulator can reach any point from any other points. So it has to know the given direct moving time of the manipulator between two arbitrary points.

According to the condition and the corresponding discussion in Section 1, the initial graph concerning spatial relation should be judged whether it is "complete graph" or not. If it is not, there would be some pairs of work points between which the manipulator can not move directly from one to another, and a new "complete graph" has to be constructed based on the initial one. The basic idea behind the construction algorithm is to construct the connected relation for those dotted pairs between which there is no direct moving path for manipulator. The weight between those dotted pair of points is the minimum moving time cost of manipulator between them. Meanwhile, the optimal path of manipulator between those two points corresponding to the minimum moving time cost is recorded. Here, algorithm Floyed in graph theory is used to find the minimum moving time cost and the optimal path of the manipulator between each discussed dotted pairs.

In the global optimal motion cycle path found by algorithm EC* SA, if there are two neighboring points which are not accessible directly to the manipulator in the initial graph, these points can be connected via the gained local optimal motion path between two points in the former construction work. But in this way, some points may be visited more than once.

3.2 With multiple configurations in each work point

To this point, multiple configurations can be chosen in each work point. Given N work points $\Theta = (C_{\Theta,1}, C_{\Theta,2}, \cdots, C_{\Theta,N})$, where $C_{\Theta,i} = (C_{\Theta,i,1}, C_{\Theta,i,2}, \cdots, C_{\Theta,i,P_{\Theta,i}})$ has corresponding $P_{\Theta,i}$ configurations ($1 \leqslant i \leqslant N$). As shown in Fig.1, an algorithm is designed to confirm the selection of configurations for each point to let each point only own a special configuration and get the global minimum time motion cycle path.

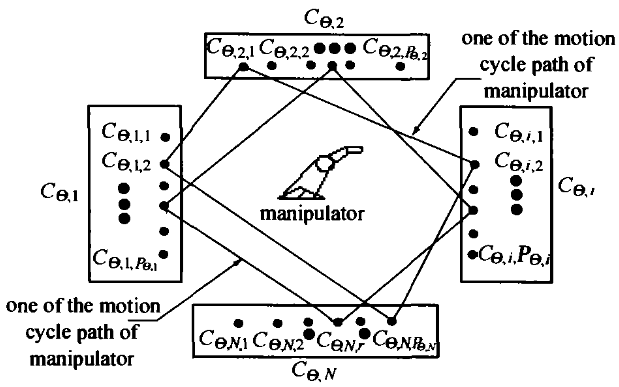

Fig. 1 Sketch map for planning path of manipulator with multiple configurations

In the algorithm EC* SA_ DP shown in the next page, the dynamic programming (DP) technique is combined to solve this problem.

Algorithm EC* SA_DP

I) Presentation of the computing method for object function.

1) Let $S_\Theta = \min\{P_{\Theta,1}, P_{\Theta,N}\}$ be the smaller of $P_{\Theta,1}$ and $P_{\Theta,N}$. Here, suppose $S_\Theta = P_{\Theta,N}$.

2) Define Min_Length_Θ = W (W is a constant of infinity).

3) for ($r = 1; r <= S_\Theta; r++$)

{

4) Here, introduce two new points $C_{\Theta,0}$ and $C_{\Theta,\hat{N}}$, where $C_{\Theta,0} = C_{\Theta,\hat{N}} = \{C_{\Theta,N,r}\}$ is the set owning the only element and their number of configurations is $P_{\Theta,0} = P_{\Theta,\hat{N}} = 1$.

5) Now, the initial order has been adapted as: $X_r = (C_{\Theta,0}, C_{\Theta,1}, \cdots, C_{\Theta,N-1}, C_{\Theta,\hat{N}})$. The number of configurations in this order is: $|X_r| = P_{\Theta,0} + P_{\Theta,1} + P_{\Theta,2} + \cdots + P_{\Theta,N-1} + P_{\Theta,\hat{N}} = \sum_{i=1}^{N-1} P_{\Theta,i} + 2$.

6) Regard X_r as a graph in which a work point configuration corresponds to a point. Let E_{X_r} be the set of pairs of points in X_r. It means that E_{X_r} is the set of edges in X_r.

7) For matrix $M_\Theta = X_r \times X_r$, time cost function $t(C_{\Theta,u,uu}, C_{\Theta,v,vv})$ denotes the minimum moving time of manipulator between the work point configuration $C_{\Theta,u,uu}$ and $C_{\Theta,v,vv}$, where $u, v = 0, 1, 2, \cdots, N-2, N-1, \hat{N}$ and $1 \leqslant uu \leqslant P_{\Theta,u}$, $1 \leqslant vv \leqslant P_{\Theta,v}$.

$t(C_{\Theta,0,1}, C_{\Theta,k,kk}) = t(C_{\Theta,N,r}, C_{\Theta,k,kk})$ and

$t(C_{\Theta,0,1}, C_{\Theta,\hat{N},1}) = t(C_{\Theta,N,r}, C_{\Theta,N,r}) = W$, where $1 \leqslant k \leqslant N$ and $1 \leqslant kk \leqslant P_{\Theta,k}$.

8) Let Path (i,j) be the shortest path from certain configuration $C_{\Theta,i,j}$ of $C_{\Theta,i}$ to $C_{\Theta,\hat{N}} = \{C_{\Theta,N,r}\}$ between which the total moving time cost is the minimum, Time_ Cost (i,j) be the total minimum moving time cost corresponding to Path (i, j), where $0 \leqslant i \leqslant N-1$ and $1 \leqslant j \leqslant P_{\Theta,i}$.

9) According to "Optimal Principle" of DP, there are two equations:

$$\begin{cases} \text{Time\_ Cost}(i,j) = \\ \min\limits_{1 \leqslant h \leqslant P_{\Theta,i+1} \text{and} (C_{\Theta,i,j}, C_{\Theta,i+1,h}) \in E_X} \{t(C_{\Theta,i,j}, C_{\Theta,i+1,h}) + \\ \text{Time\_ Cost}(i+1,h)\} \text{ if } 0 \leqslant i \leqslant N-2, \\ \text{Time\_ Cost}(N-1,j) = \\ t(C_{\Theta,N-1,j}, C_{\Theta,\hat{N},1}) = t(C_{\Theta,N-1,j}, C_{\Theta,N,r}) \end{cases}$$

10) Let $D(i,j) = h$ be the corresponding h which makes the right of the above first equation get the minimum value.

11) At last, for the initial work points order, Time_ Cost$(0,1)$ becomes the total minimum moving time cost via solving the above DP equation. Let Length_ X_r = Time_ Cost$(0,1)$ be the final result.

12) if (Length_ X_r < Min_ Length_ Θ)

{

13) Min_ Length_ Θ = Length_ X_r;

14) Do the recoding: Path_ Θ = $(C_{\Theta,0,q(0)}, C_{\Theta,1,q(1)}, C_{\Theta,2,q(2)}, \cdots, C_{\Theta,N-1,q(N-1)}, C_{\Theta,\hat{N},q(\hat{N})})$.

where $q(0) = 1, q(\hat{N}) = 1$, and $q(i) = D(i-1, q(i-1))(1 \leqslant i \leqslant N-1)$.

$C_{\Theta,0,q(0)} = C_{\Theta,0,1} = C_{\Theta,\hat{N},q(\hat{N})} = C_{\Theta,\hat{N},1} = C_{\Theta,N,r}$.

}

}

The object function of Θ is: $f(\Theta)$ = Min_ Length_ Θ. The corresponding optimal cycle path is Path_ Θ.

II) Getting the optimal solution by means of the similar process of algorithm EC* SA.

The following process is similar to the process of algorithm EC* SA. The unique difference is the computing method of corresponding object function.

4 Numerical simulation results

C++ programs have been written to implement the above two algorithms. The developed platform is Visual C++ 6.0 and the computer used here is Intel Pentium 4 (1.6GHz) + 256 MB RAM.

Some points in the workspace of robot are generated randomly. In the case, 76 tests from $N = 5$ up to 80 work points with one test for each value of N are generated. Fig. 2 is the comparison diagram of CPU time for each test after the convergence of several algorithms. The general performance of EC* SA is the best. For example, when $N = 80$, ENM is 6.4106 times longer than EC* SA. Although its CPU time is longer than SGA, it can get more high-quality solutions. Fig. 3 makes a comparison of EC* SA and SGA in convergence curve of minimum time considering 80 points. Here, the minimum times obtained via SGA and EC* SA are 37.0453s and 34.4637s respectively.

4.1 Numerical simulation for algorithm EC* SA

The test robot prototype is a 6 – DOF YASAKAWA UP6 MOTOMAN robot. In terms of large numbers of work points, the running results of algorithm EC* SA are compared with elastic net method (ENM)[5], simulated annealing (SA), and simple genetic algorithm (SGA), as shown in Fig.2 and Fig.3.

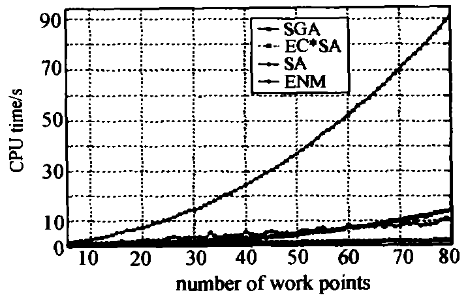

Fig. 2 Comparison diagram of convergence time considering 5 to 80 work points

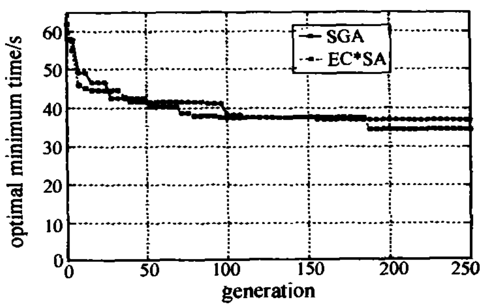

Fig. 3 Comparison between EC* SA and SGA in convergence curve of minimum time considering 80 work points

4.2 Numerical simulation for algorithm EC* SA_DP

The same example as in [5] is considered. For a 2-DOF robot (one prismatic joint and one rotational joint), there are 15 work points to be planned (there were only 10 points scheduled in [5]). In the workspace, each point can be reached with two different configurations ("elbow right" and "elbow left"). Joint limits, velocity and acceleration parameters on each link are given in Table 1. And the detailed coordinates are given in Table 2. One of the optimal results obtained by EC* SA_DP is shown in Fig. 4. The corresponding minimum time is 15.8099 s.

Table 1 Kinematics parameters

joint Limits	velocity	acceleration
$q_1 \in [-50,140]$/mm	$K_{v1} = 20.67/s$	$K_{a1} = 183.34/s^2$
$q_2 \in [-10,240]$/(°)	$K_{v2} = 51.56/s$	$K_{a2} = 103.12/s^2$

Table 2 Joint coordinates of 15 work points

No.	q_{11}	q_{21}	q_{12}	q_{22}
1	-28.7	14.5	48.7	165.5
2	13.5	48.6	66.4	131.4
3	41.3	14.5	118.7	165.5
4	65.3	30.0	134.6	150.0
5	30.3	7.2	99.7	172.8
6	-6.4	48.6	46.4	131.4
7	-4.6	30.0	64.6	150.0
8	33.5	48.6	86.4	131.4
9	50.0	0.0	130.0	180.0
10	-40.0	0.0	40.0	180.0
11	-46.0	22.5	27.9	157.5
12	14.6	-10.0	93.4	190.0
13	81.1	50.0	132.5	130.0
14	41.6	45.0	98.2	135.0
15	-1.48	30.0	67.8	150.0

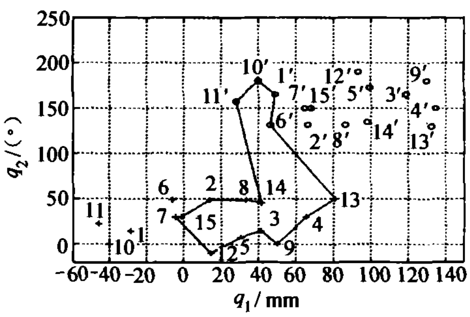

Fig. 4 One of optimal results obtained by EC* SA_DP

5 Virtual implementations on PVR-based control platform

In this section, the presented algorithms are implemented on a PVR-based control platform developed by the authors and a useful metaphor called "virtual instructional moving path" is gotten. The function can be described via an example as follows. When a user wants to move a robot manipulator, he can move the arm on the PVR-based control platform along an optimized moving path got by the proposed algorithm in the virtual environment at first. As soon as the user finishes moving one step along the virtual path, the corresponding practical movement of the robot manipulator will be carried out in the practical environment with the help of "task deduction" and "action planning" modules[10, 11]. The graphic simulation results are shown in Fig.5 and Fig.6 respectively. In the figures, the big black dots denote the prescribed work points that have to be visited by the manipulator.

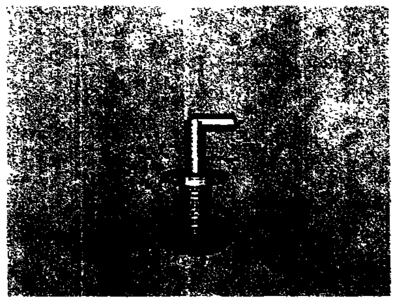

Fig. 5 One of middle states of manipulator on planned virtual instructional moving path

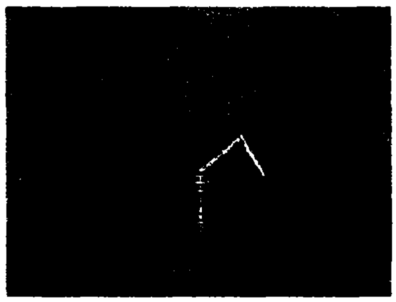

Fig. 6 Starting state of manipulator on planned virtual instructional moving path

6 Conclusions

MTMPP of robot manipulators to execute some relatively complex tasks is a rather difficult optimal problem. Some approaches were proposed in solving PTP task. However, they could not get over the main difficulty of solving MTMPP problem. Aiming at those essential difficulties, this paper has made the following contributions.

1) For the instance of high-order DOF robot manipulator visiting large numbers of work points with unique or multiple configurations, two novel algorithms EC* SA and EC* SA_ DP are presented, and near optimal solutions are obtained.

2) Virtual simulation is carried out on the PVR-based control platform for the first time, and some useful virtual simulation results are obtained. It could be expanded as the metaphor operations in the virtual world and be used in the field of teleoperation as well.

References:

[1] BOBROW J E, DUBOWSKY S, GIBSON J S. Time-optimal control of robotic manipulators along specified paths [J]. *Int J of Robotics Research*, 1985, 4(3): 3 - 17.

[2] MACJEJEWSKI A A, FOX J J. Path planning and the topology of configuration space [J]. *IEEE Trans on Robotics and Automation*, 1993, 9(4): 444 - 456.

[3] SHIN K G, MCKAY N D. Selection of near minimal time geometric paths for robotic manipulators [J]. *IEEE Trans on Automatic Control*, 1986, 31(6): 501 - 511.

[4] DUBOWSKY S, BLUBAUGH T D. Planning time-optimal robotic manipulator motions and work places for point-to-point tasks [J]. *IEEE Trans on Robotics and Automation*, 1989, 5(3): 377 - 381.

[5] PETIOT J F, CHEDMAIL P, HASCOET J Y. Contribution to the scheduling of trajectory in robotics [J]. *Robotics and Computer-Integrated Manufacturing*, 1998, 14(3): 237 - 251.

[6] FREUND E, ROSSMANN J. Projective virtual reality: bridging the gap between virtual reality and robotics [J]. *IEEE Trans on Robotics and Automation*, 1999, 15(3): 411 - 422.

[7] LUO Xiong, FAN Xiaoping. Simulated annealing algorithm for MTSP [J]. *J of Central South University of Technology*, 2001, 32 (Spec.1): 115 - 118.
(罗熊，樊晓平. MTSP的模拟退火算法[J]. 中南工业大学学报, 2001, 32(专辑): 115 - 118.)

[8] WANG Xiaoping, CAO Liming. *Genetic Algorithms: Theory, Applications and Software* [M]. Xi'an: Xi'an Jiaotong University Press, 2002.
(王小平，曹立明. 遗传算法——理论、应用与软件实现[M]. 西安：西安交通大学版社, 2002.)

[9] WANG L. *Intelligent Optimization Algorithms with Applications* [M]. Beijing: Tsinghua University Press; New York: Springer-Verlag, 2001.

[10] LUO Xiong, FAN Xiaoping. Principle and implementation for projective virtual reality with its applications in robot control [J]. *High Technology Letters*, 2002, 12(3): 96 - 100.
(罗熊，樊晓平. 投射式虚拟现实的原理与实现及其在机器人控制中的应用[J]. 高技术通讯, 2002, 12(3): 96 - 100.)

[11] LUO Xiong, FAN Xiaoping, CHEN Songqiao, et al. MTTP for robot manipulators based on intensified evolutionary programming and its application in PVR environment [A]. *Proc of the 4th World Congress on Intelligent Control and Automation* [C]. Shanghai: Press of East China University of Science and Technology, 2002, 1: 26 - 31.

作者简介:

罗　熊　(1976—),男,系统分析员(高级工程师),IEEE机器人和自动化学会会员,现为中南大学信息科学与工程学院博士研究生,主要研究方向为机器人控制,虚拟现实技术,计算机网络等,曾获2002年度IEEE控制系统学会北京分会颁发的优秀青年论文奖,已发表有关论文20余篇,E-mail: xiongluo@sohu.com;

樊晓平　(1961—),男,博士,教授,博士生导师,主要研究方向为机器人控制,虚拟现实技术,智能交通系统等,已发表有关论文80余篇.