Niche Pseudo-Parallel Genetic Algorithms for Path Optimization of Autonomous Mobile Robot - A Specific Application of TSP

Zhihua Shen and Yingkai Zhao
Faculty of Engineering, The University of Melbourne
Faculty of Automation, The Nanjing University of Technology

1. Introduction

Path planning of autonomous mobile robot is pivotal technique for machine intelligence, which aims to find a non-collision path from initial position to objective position according to evaluation functions in an obstacle space[1]. It can be described as traveler salesman problem (TSP), a typical combination optimization problem, which belongs to the well-known NP-hard optimization[2]. The mathematical definition can be regarded as a map G = (V, E), where each line e∈E has a nonnegative power ω(e). The aim is to find out a Hamilton circle noted with C of map G in order to obtain a minimum power $W(C)=\sum_{e\in E\ (C)}\omega(e)$.

Some traditional methods such as greed arithmetic, vicinage arithmetic and dynamic programming algorithm[3] do not behave a good performance on combination explosion aroused by rapid increment in exponent within a solution set of mathematic model, also known as the very quick accretion in both aspects of space and time complication along with the increase of degrees. A very promising direction is the genetic algorithm (GA) except for the traditional methods. Genetic algorithm is numerical optimization method[4] based on the theory of genetics and natural selection. It is a generally probabilistic and adaptable concept for problem solving, especially suitable for solving difficulty optimization and evolution problems, where traditional methods are less efficient.

An advanced genetic algorithm, niche pseudo-parallel genetic algorithm (NPPGA) is presented based on simple genetic algorithm (SGA), niche genetic algorithm (NGA) and parallel genetic algorithm (PGA) to further improve the GA for robot path optimization. Research on NPPGA is available for lots of practical problems such as path routing optimization, nets organization, job distribution, scheduling optimization etc.

2. Mechanism of niche pseudo-parallel genetic algorithms

The foundation of NPPGA is genetic algorithm, which is a class of global, adaptable, and probabilistic search optimization and revolution algorithm gleaned from the model of organic evolution and also simulates the genetics and evolution of biologic population in nature. GA adopts naturally evolutionary model such as selection, crossover, mutation, deletion and transference. Mathematically, this evolutionary process is a typical algorithm to find out the optimal solution via iteration search among multi-element in a NP set. As an optimal method applied with biologic genetics and evolutionary mechanism [5], GA totally

Source: Travelling Salesman Problem, Book edited by: Federico Greco, ISBN 978-953-7619-10-7, pp. 202, September 2008, I-Tech, Vienna, Austria

embodies a classical biologically evolutionary theory depicted as natural selection. Simple genetic algorithm can be defined as *SGA= (M, C, F, M_o, P_s, P_c, P_m, T)* [3], where *C* is a fixed bit-string code, *F* is a fitness evaluation function, M_o is an initial population of biologic colony and P_s, P_c, P_m are probabilities of selection, crossover and mutation respectively.

2.1 Proposal of NPPGA

Theoretically, genetic algorithm is able to trace on the optimal solution by a stochastic method on the sense of probability. On the contrary, GA has some prominent problems in practical application such as premature convergence, feebleness in local search, low rate of convergence etc. A simplex renewal from one population to another is hard to keep population diverse and avoid premature convergence.

Simple genetic algorithm is totally a stochastic method, which aims to settle with the problem where several different individuals are required to optimize in a cryptic and parallel process [6]. However, the rate of evolutionary process is still lower because of its essentially serial mechanism. In addition, before tracing on the globally optimal solution, the SGA may converge to a local one, which causes population trend to un-animousness and results in premature. To further improve the GA and avoid these disadvantages, we firstly divided original population into several groups known as pseudo- parallel operation to accelerate the rate of genetic algorithm computation and maintain the population diversity in order to reduce the rate of premature simultaneously. Based on the former step, niche genetic mechanism is introduced into pseudo-parallel genetic algorithm to further restrain the premature phenomenon. A method based on sharing functions is proposed to transfer genetic information to keep population diversity and avert from rapid increment of some special individuals, in other words, we created several niches among the population by pseudo-parallel technique to complete the process both of local and global solution optimization.

2.2 Pseudo code and layered flow chart of NPPGA

The Pseudo code of NPPGA is showed as following.

```
Begin s:=0;
initialize $P(0) := \{x_1(0),...,x_m(0)\} \in \Re^m$ where $\Re = \{0,1\}^l$;
evaluate $P(0) := \{\Phi(x_1(0)),...,\Phi(x_m(0))\}$ where $\Phi(x_k(0)) = \delta(f(\gamma(x_k(0))), P(0))$;
while $(\Im P(t) \neq true)$ do t:=0; $P(t) := \{P_1(t), P_2(t),...,P_i(t),...,P_n(t)\}$;
    while $(\Im P(i) \neq true)$ do
        recombine: $x_k^{'}(t) := r_{\{p_c,z\}}^{'}(P_i(t))\ \forall k \in \{1,...,m/n\}$;
        mutate: $x_k^{''}(t) := m_{\{p_m\}}^{'}(x_k^{'})\ \forall k \in \{1,...,m/n\}$
        niche operator: $x_k^{'''}(t) := \wp_{\{\gamma\}}^{'}(x_k^{''}) \forall k \in \{1,...,m/n\}$ with formula(1) and (2);
        delete: $x_k^{''''}(t) := d_{\{\alpha\}}^{'}(x_k^{''''}) \forall k \in \{1,...,m/n\}$;
        evaluate: $P_i^{'''}(t) := \{x_i^{''''}(t),...,x_{m/n}^{''''}(t)\} \in \Re^{m/n}$; $\{\Phi_i(x_1^{''''}(t)),...,\Phi_i(x_{m/n}^{''''}(t))\}$
                where $\Phi_i(x_k^{''''}(t)) = \delta(f(\gamma(x_k^{''''}(t))), P_i(t-\varpi))$;
                select: $P_i(t+1) := s(P_i^{''''}(t))$
        where $P_s(x_k^{''}(t)) = \Phi_i(x_k^{''''}(t)) / \sum_{j=1}^{m/n} \Phi_i(x_k^{''''}(t))$;
        t:=t+1; end
information exchange: $P(s+1) = c\{P(s),\ P_i^{''''}(t)\}$; s=s+1; end
```

Fig. 1 describes the layered structure of NPPGA by a flow chart.

Fig. 1 Pseudo code of NPPGA

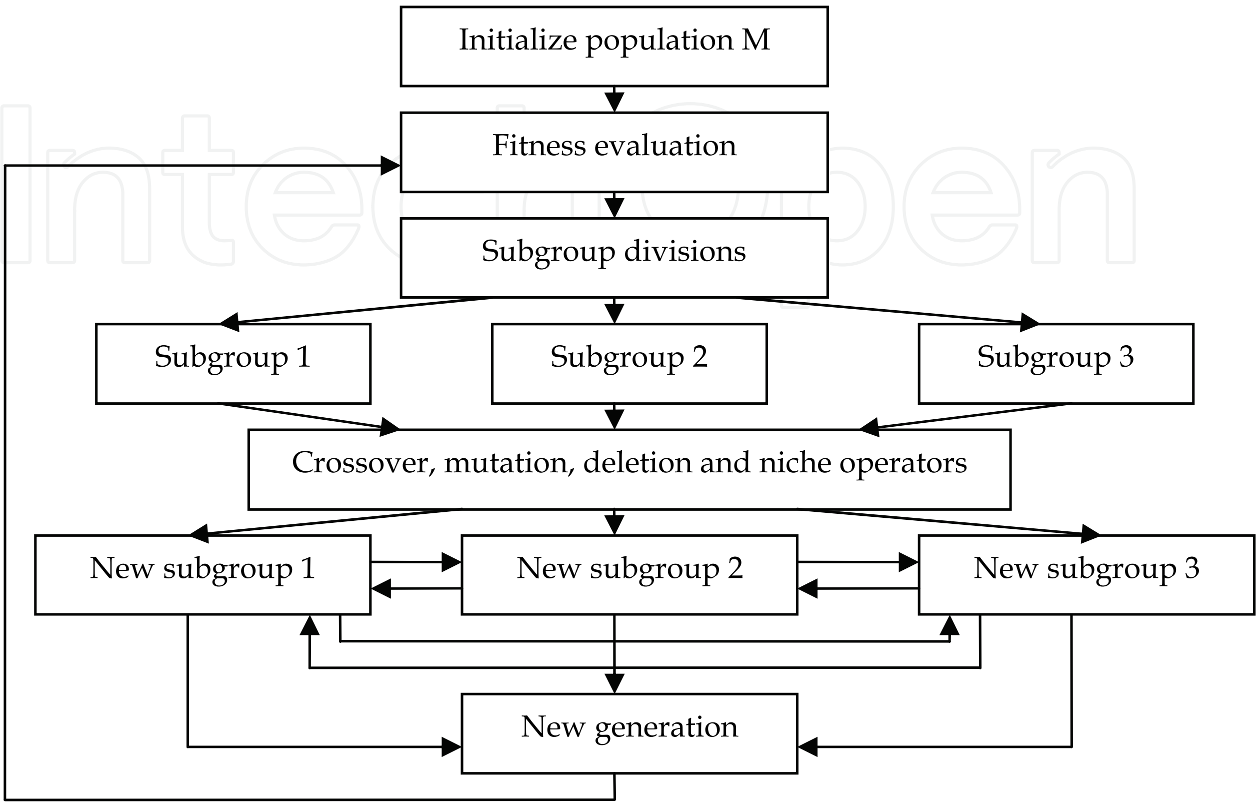


Fig. 1. Flow chart of NPPGA

2.3 Arithmetic description of NPPGA

Each step in niche pseudo-parallel genetic algorithms is demonstrated as following based on niche genetic algorithm and parallel genetic algorithm [3] [6]:

1. Initialize the genetic counter $t \leftarrow 0$;
2. Generate original population $P(t)$ made up of initial individuals which are divided into several subgroups in the form of $P(t)=\{P_1(t), P_2(t),\ldots,P_i(t),\ldots, P_n(t)\}$, where n is a integral number of subgroup.
3. Calculate each individual fitness function $F_j(j=1,2,\ldots,m/n)$ in every subgroup $P_i(t)(i=1,2,\ldots,n)$ of the population;
4. Execute independent evolution among each group $P(t)=\{P_1(t), P_2(t),\ldots,P_i(t),\ldots, P_n(t)\}$;
   i. Independently initialize evolutionary counter $s \leftarrow 1$ and select a subgroup fitness function to give an evaluation of each individual;
   ii. Make a lower taxis upon the individual fitness and storage q individuals in the former sequence where $q<m/n$;
   iii. Reproduce among $P_i(t)$ by the operator P_s in form of $P_i^{'}(t) \leftarrow selection[P_i(t)]$;
   iv. Crossover among $P_i^{'}(t)$ by the operator P_c in form of $P_i^{''}(t) \leftarrow crossover[P_i^{'}(t)]$;
   v. Mutate among $P_i^{''}(t)$ by the operator P_m in form of $P_i^{'''}(t) \leftarrow mutation\ [P_i^{''}(t)]$;
   vi. Delete among $P_i^{'''}(t)$ by the operator γ in form of $P_i^{''''}(t) \leftarrow deletion[P_i^{'''}(t)]$;

vii. Use a niche operator, which needs to combine $(m/n)-\gamma F[P_i^{'''}(t)]$ individuals in the subgroup $P_i^{''''}(t)$ with the former q excellent individuals that are saved early into a new population M_i including $(m/n)+q-\gamma F[P_i^{'''}(t)]$ individuals, to wash out some inferior ones. When $\| X_i - X_j \| < L$, a Hamming distance is computed by

$$\| X_i - X_j \| = \sqrt{\sum_{k=1}^{m/n} (x_{ik} - x_{jk})^2}, \quad i = 1,2,...m/n+q-1,\ j = i+1,...m/n+q \tag{1}$$

L is the distance between contiguous generations. A penalty function $F_{\min(x_i,x_j)} = Penalty$ is used after comparing fitness between X_i and X_j. The penalty criterion is

$$F_i^{'}(X) = \begin{cases} F_i(X) & (if\,(F_i(X) \geq F_j(X))\&\,\|(X_i - X_j\| < L)) \\ F_i(X) - P(X) & (if\,(F_i(X) < F_j(X))\&\,\|(X_i - X_j\| < L)) \end{cases} \tag{2}$$

viii. Realign in a lower sequence according to each fitness of (m/n)+q individuals and store former q individuals again;

ix. End niche heredity if evolutionary results fit with ending conditions, or renew independently evolutionary counter s←s+1 and turn to step iii while generating m/n individuals into next generation in step vii;

5. Transfer information usually with stepping-stone model, island model and neighborhood model among *Pi(t)(i=1,2,…,n)* to obtain the next generation $P(i+1) \leftarrow exchange\ [P(t), P_i^{'''}(t)]$;
6. End parallel heredity when evolutionary results fit with ending conditions, otherwise, renew independently evolutionary counter *t←t+1* and turn to step (4).

3. Robot path optimization by NPPGA

It is well known that the problem of "Robot touring around Pekin" is typically practical application of TSP. Based on discussion in section 2, mechanism of niche pseudo-parallel genetic algorithm is investigated. In this section, NPPGA is used to solve the traveling salesman problem especially in the model of path optimization of robots. Each individual code is described in bit-strings of fixed length 18, which stands for paths between each two cities. Then an entire serial named chromosome $\vec{X}_i = (X_{i,1}, X_{i,2}, ..., X_{i,m})$ can be obtained in each individual space $S = \{X_{i,1}, X_{i,2}, ..., X_{i,m}\}^m$ that belongs to subgroup space $S^{m/n}$. The selection operator that is known as survival probabilities in solving path optimization is in the canonical form [6]

$$p_s = P\{T_s^{\alpha}(\vec{X}) = X_i\} = \frac{f^{\alpha}(X_i)}{\sum_{k=1}^{m/n} f^{\alpha}(X_k)} \tag{3}$$

To optimize the robotic paths, crossover operator, emphasized as the most important search operator of genetic algorithm, is introduce by[3]

$$P\{T_c(Y_{i,1}^k, Y_{i,2}^k) = X_k^{'}(n+1)\} = \begin{cases} {kp_c}/{m}, X_k^{'}(n+1) \neq Y_{i,1}^k and X_k^{'}(n+1) = AY_{i,1}^k + (I-A)Y_{i,2}^k \\ (1-p_c) + {kp_c}/{m}, X_k^{'}(n+1) = Y_{i,1}^k \\ 0, other \end{cases} \tag{4}$$

Small mutation and deletion rates are also used in solving this problem to guarantee that each individual do not differ genetically very much from its ancestor. In other words, it keeps the diversity of path space even though local convergence exists. Niche operator is demonstrated in Eqs. (1) and (2). Furthermore, we exchange different information of excellent path serials among subgroups based on islands model. All these parameters used in NPPGA are showed in table 1.

Parameters	Used in NPPGA
Selection rate	α=1
Mutation operator	P_m=0.0015
Recombination operator	P_r={0.72,4}
Deletion operator	γ=0.0027
Niche operator	Hamming Distance
Length per object variable	L=18
Population size	50

Table 1. Parameters of NPPGA

In the experiment, a single step NPPGA is used to solving the problem of path optimization and evolution of "Robot Tour". 8 optimal solutions can be obtained shown in table 2. The length of optimal path has been changed into standard units where (Remnant of optimal paths)= (length of paths)-(shortest distance) and (Ratio of relative paths) =(remnant)/(shortest distance).

Path	Path 1	Path 2	Path 3	Path 4	Path 5	Path 6	Path 7	Path 8
Length of optimal paths	96.17 (NPPGA)	96.79	97.20	97.59	98.48	98.75	102.17	102.77
Remnant of optimal paths	0 (NPPGA)	0.62	1.03	1.42	2.31	2.58	6.00	6.60
Ratio of relative path	0.26% (SGA)	0.86%	1.54%	2.2%	2.58%	3.73%	5.80%	6.92%
	0.18% (DPGA)	0.70%	1.32%	1.80%	1.93%	3.12%	5.95%	6.90%
	0% (NPPGA)	0.640%	1.071%	1.477%	2.042%	2.683%	6.239%	6.863%
Computation complexity	SGA	346 generations and 20760 count steps						
	DPGA	294 generations and 19500 count steps						
	NPPGA	276 generations and 16560 count steps						

Table 2. Experimental results in path optimization by single step NPPGA

According to experimental data, we illustrated an evolutionary process by niche pseudo-parallel genetic algorithm in figure 2. Compared with SGA and DPGA, the performance conducted by NPPGA is better. The computation complexity of NPPGA is 16560 count steps within 276 generations while SGA and DPGA are 20760 and 19500 respectively. Global optimization path other than local solution can be achieved by NPPGA when generations approach less than 300. Simultaneously, a remnant comparison is shown in Fig. 3. Although NPPGA has the peak error for some individual evolutionary processes caused by stochastic researching, it perform a lowest remnant error to the optimal path while the remnant of DPGA is a little bit large than NPPGA. The shortest route can be described in the following serial.

"start→dong_wu_yuan→zhong_guan_cun→yuan_ming_yuan→yi_he_yuan→xiang_shan→shi_san_ling→ba_da_ling→yong_he_gong→bei_hai_gong_yuan→gu_gong→tian_an_men→wang_fu_jing→beijing_zhan→tian_tan→shi_jie_gong_yuan→xi_dan→shi_ji_dan"

The total distance is 96.17 in standard units and actually shorter in practical robot tour.

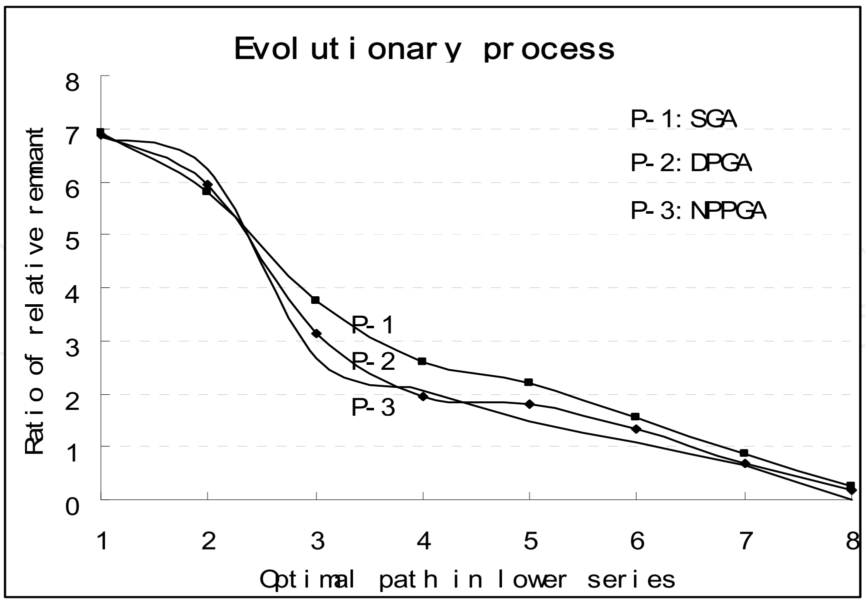


Fig. 2. Evolutionary process of NPPGA

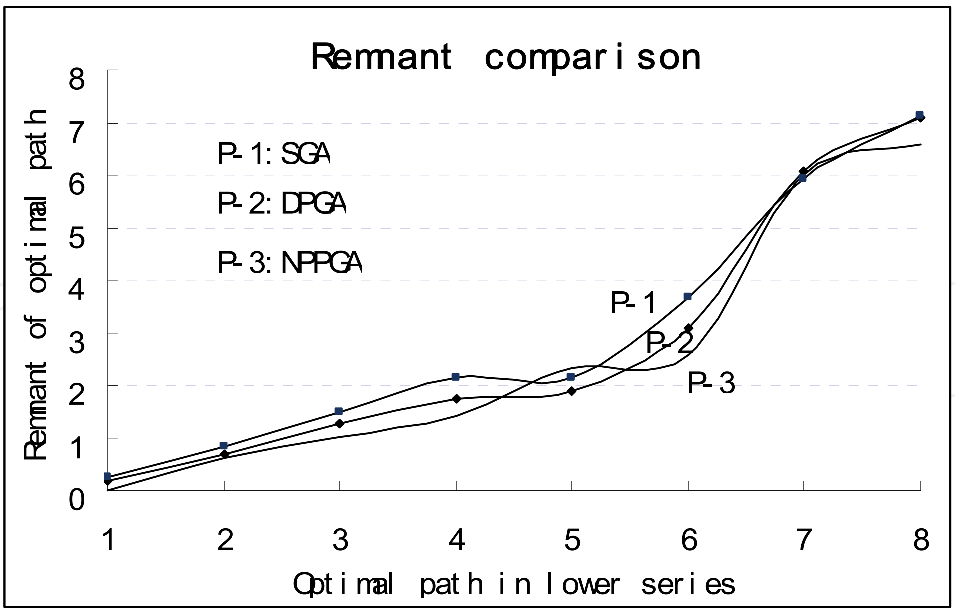


Fig. 3. Remnant comparison of NPPGA

4. Conclusion

The research, based on combination of niche genetic algorithm and pseudo parallel genetic algorithm, comes into being NPPGA technique which both considers the rate of genetic evolution and diversity of population. The strategy seems to be able to restrain the premature among population and closely cooperate with each other to improve the overall search performance. We presented NPPGA and used a single step NPPGA to figure out the optimal paths in “Robot tour around Pekin”, which is a practical application of traveling salesman problem. Experiments show that the optimal route can be obtained. We believe that NPPGA and other advanced GAs will become a robust tool for path optimization and other potential applications.

5. References

Hu Y R, Yang S X. *Knowledge based genetic algorithm for path planning of a mobile robot* [A]. Proc. Of 2004 Robotics and Automation, IEEE international conference ICRA '04 [C]. New Orleans, USA, 2004, V(5):4350-4355.

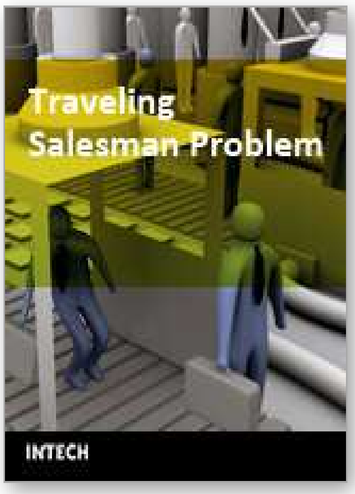
Daoxiong Gong, Xiaogang Ruan. *A hybrid approach of GA and ACO for TSP [A]*. Intelligent Control and Automation, WCICA 2004. Fifth World Congress [C]. Hangzhou, China, 2004 V(3):2068– 2072.

Zhe Lei-lei, Chen Huan-yang. Mathematical Foundation of Computer Intelligence [M]. Science Publishing Company, Beijing, 2002, P:112-142.

Pullan, W. *Adapting the genetic algorithms to the traveling salesman problem [A]*. Evolutionary Computation CEC '03 Congress [C]. Canberra, Australia, 2003, 2(8):1029 – 1035.

Huai-Kuang Tsai, Jinn-Moon Yang, Yuan-Fang Tsai *et. al. An evolutionary algorithm for large traveling salesman problems [J]*. Systems, Man and Cybernetics, IEEE Transactions, 2004, 34(4):1718 – 1729.

Xiong Shengwu, Li Chengjun. *A distributed genetic algorithm to TSP [A]*. Intelligent Control and Automation Proceedings of the 4th World Congress [C]. Shanghai, China, 2002, 3(10): 1827-1830, 2002



Traveling Salesman Problem
Edited by Federico Greco

ISBN 978-953-7619-10-7
Hard cover, 202 pages
Publisher InTech
Published online 01, September, 2008
Published in print edition September, 2008

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821