Motion Generation of Multi-Legged Robot in Complex Terrains by using Estimation of Distribution Algorithm

Min JIANG, Zhongqiang HUANG, Guiying JIANG, Minghui SHI, and Xiangxiang ZENG

***Abstract*—Motion generation is one of the most important and challenging problems in multi-legged robot research. Most of the existing methods show a good fulfillment of the requirements of robots in structured environments. However, it still faces many challenges to generate motions effectively and quickly for multi-legged robot works in complex environments. In this paper, we put forward a method which converts the motion generation problem into a Multi-objective Optimization Problem (MOP), which will make the robot not only run as fast as possible, but also save energy, and then use a distribution estimation algorithm, the trend prediction model method, to obtain motions for a six-legged robot. Experiments show that this method is effective.**

***Index Terms*—Robotics, Multi-legged Robot, Motion Generation, Estimation of Distribution Algorithm.**

I. Introduction

Due to its excellent balance ability and load capacity, multi-legged robot has played an extremely important role in disaster relief, exploration and other fields. In the study of multi-legged robot, it is still a crucial and challenging problem that how to generate motions for a robot in the complex environment accurately and rapidly. One main reason is the case that compared with other types of robots, the multi-legged robot has more Degrees of Freedoms (DoF), so it is more costly, complicated to generate appropriate motions.

In essence, the motion generation of multi-legged robot can be formulated as a optimization problem. For a considerable period, analytical methods occupied an important position for solving the problem. Generally, analytical methods often build a model for the motion generation problem and then solve the optimization problems behind the model via analytical approaches to obtain corresponding motions for the robot. The advantages of the analytical methods are precise, and the speed of motion generation is fast, but the deficiency is also obvious. First, the generalization abilities of the model are usually not strong, so the application scopes of the methods are limited. Secondly, it is very difficult to establish an accurate model for different types of robots and the environments they are working in. Therefore, more and more researchers [1] in this field pay attention on the methods based on computational intelligence or machine learning approaches.

For example, in [2], the authors proposed using genetic algorithm to generate motions for the four-legged robot. In [3], the author proposed a genetic-fuzzy system to help a six-legged robot to find a collision-free, time minimal path and to plan its stable gait. In [4], motion generation of multi-legged robot by using evolutionary approach and neural network is reported. Evolution computation method is appropriate for the gait generation problem. Genetic algorithm(GA) is the technique that most often used in gait optimization problem [5] [6] [7]. Steady state genetic algorithm is applied for generating motion sequences of six-legged robot in [8]. Gait motion acquistion and pass planning of robot by using reinforcement learning is studied in [9]. A method for motion pattern generation of multi-legged robot by using spiking neural network(SNN) in [10]. The study which combines genetic algorithm with reinforcement learning to acquire walking behavior is discussed in [11]. A policy gradient reinforcement learning approach for generating a fast walk on a legged robot is proposed in. In [12], the authors proposed a method that sequences and patterns of motion are formed by using minimal generation gap and a genetic algorithm. However, the above techniques based ib the traditional computational intelligence methods are still to find a decent solution of gait optimization problem [1].

Unlike evolutionary strategy based on individual, Estimation of Distribution Algorithm (EDA) [13] combines evolutionary strategy based on group and probability and statistics, so as to make the EDA can effectively deal with complex optimization problems, and the key characteristic of the EDAs is that it searches the solution space by repeatedly modeling and sampling the space of the candidate solutions. Owing to the optimization process of EDA is a probability distribution of updated repeatedly, make it compared with the traditional evolutionary algorithm has the following advantages. At first, the adaptive operator of the EDA can be adjusted according to the problem to be solved operator. Secondly, The EDA normally has clear problem structure, and the structure provides the concrete probability model of the problem solution, which makes the analysis and information mining of the model easy. Thirdly, the EDA simply records the probabilistic model parameters without directly preserving the population, reducing the need for memory. These features make the EDA algorithm suitable for handling motion generation problems.

The contribution of this article is that we convert the motion generation problem of a six-legged robot which works in the rough terrains into a Multi-objective Optimization Problem (MOP) [14], or to describe it more accurately, We hope the

Min JIANG and Minghui SHI are with the Department of Cognitive Science and Fujian Key Laboratory of Brain-inspired Computing Technique and Applications (Xiamen University). Xiangxiang ZENG is with the Department of Computer Science of Xiamen University. Guiying JIANG is with the College of Foreign Languages of Xiamen University. Zhongqiang HUANG is an algorithm engineer of Sangfor Technologies Company.

The Corresponding author: Xiangxiang Zeng, email: xzeng@xmu.edu.cn

978-1-5386-2726-6/17/$31.00 ©2017 IEEE

six-legged robot will not only run as fast as possible, but also save energy. So we use two performance indices, Energy Stability Margin [15] and Froude number [16], to evaluate the potential motions, and then use an Estimation of Distribution Algorithm, the Trend Prediction Model Based Multi-Objective Estimation of Distribution Algorithm (TPN-EDA) [17], to solve the proposed optimization function and then obtain the appropriate motions.

The rest of this paper is organized as follows. In Section II, we briefly introduce background and the robot model used in this research. In Section III, we describe how to use the TPM-EDA to obtain motions for a six-legged robot in the rough terriers. In Section IV, the experimental results and some discussions are presented. In the Section V, we sum up the main work of this research and future research direction is also highlighted.

II. Preliminaries

A. Estimation of Distribution Algorithms

The main feature of EDAs [13] is that this algorithm incorporates probabilistic modeling into the evolutionary framework for optimization. Instead of individual-based evolution strategy, EDAs use a population-base strategy to evolve the whole population and explore the search space through building and sampling probabilistic models of promising candidate solutions.

Compared to traditional Evolutionary Algorithms (EAs), EDAs possess some unique advantages: 1) they provide explicit probabilistic models, so we can analyze these models mathematically; 2) a prior knowledge can be incorporated into the models in a more principled way; 3) instead of managing a population through variations, reducing memory requirement in EDAs is highly expected. A majority of these designs would generally involve the following two steps:

1) Firstly, a probabilistic model is built for modelling the solution space, usually through the process of population evaluating, candidate selection, and statistical machine learning. We call this step as *Model Building*.
2) Based on the constructed model, some kind of sampling methods is used to generate new population. We call this step as *Sampling*.

One of the key questions for designing a successful EDA is to find a way to depict the distribution of the solutions correctly, and the most common method is to build a probabilistic model to achieve the goal.

B. Performance Indices for Multi-legged Robots

Compared to traditional wheeled robots, multi-legged robots have more choices to maintain static balance, so many research works for the multi-legged robots have concentrated on motion generation or gait planning in a complex environment.

In multi-legged robot community, the term support pattern is often used for support polygon. By omitting the inertial effects result from body and leg acceleration, and if a control algorithm can make the projection of the Center of Mass (CoM) fall inside the support polygon, the robot can maintain its balance.

For a given configuration of a walking multi-legged robot, the stability margin $S_m = min(d_1, d_2, d_3)$, where d_1, d_2 and d_3 are the distance of the vertical projection of the center of mass to the boundaries of the support polygon in the horizontal plane. Fig. 1. illustrates the support polygen and the stability margin.

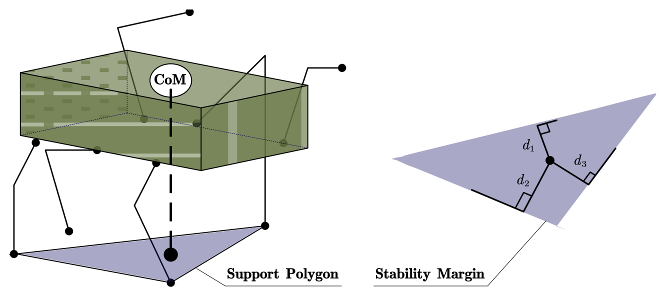

Fig. 1: Support Polygon and Stability Margin

Messuri et.al [15] proposed the Energy Stability Margin (ESM) to tumble the robot around the edges of the support polygon as the minimum potential energy required, and its definition are given as follows:

$$S_{ESM} = \overset{l_s}{\underset{i}{\min}}\,(mgh_i)\,. \tag{1}$$

Where i denotes the segment of the support polygon considered the rotation axis, g is acceleration of gravity and h_i is the variation of the Center of Gravity (CG) height during the tumble, and l_s is the number of supporting legs. The formal definition of the h_i comes from:

$$h_i = |R_i| \cos\psi(1 - \cos\theta),$$

where R_i is the distance from the CG to the rotation axis, ψ is the inclination angle of the rotation axis relative to the horizontal plane, and θ is the angle that R_i forms with the vertical plane.

In the field of fluid mechanics, the Froude number is used to explain the behavior of surface waves. Since both of surface waves and legged locomotion are dynamic motion in gravity, in [16], the authors used it to evaluate animal locomotion, and we can calculate a Froude number by the following equation:

$$F_{r2} = \frac{V}{gh}, \tag{2}$$

where V is the walking speed, g is the acceleration due to gravity, and h is the height of hip joint from the ground. The Froude number always be take as a dimensionless speed for legged robots.

C. The Sixed-legged Robot

In this work, a six-legged robot (shown in Fig. 2) is used as the experimental platform. The legs are identified by number from 1 to 6. The leg #1 and leg #2 are treated as the front of robot and the leg #5 and the leg #6 are the rear of it. The legs of the robot are placed outside of the main body vertically in equal intervals. There are three joints in each leg, so the robot possesses 18 degrees of freedom. All legs move independently at the same time during a cycle of simulation. The posture of the robot can be calculated by the forward kinematics equation, which depends on the position of each foot. Furthermore, the position of the foot is determined by the joint rotation angle and length of each part of the leg as showed in Fig 3, where θ_1 represents rotation angle around y axis and θ_2 and θ_3 is the rotation angle around x axis. The reach position $P = (p_x, p_y, p_z)$ of the leg can be calculated by following coordinated transformation equation:

$$\begin{aligned}\mathbf{P} = \mathbf{Trans}(0, l_0, 0) \cdot \mathbf{Rot}(y, \theta_{m,1}) \cdot \mathbf{Trans}(0, l_1, 0) \\ \cdot \mathbf{Rot}(x, \theta_{m,2}) \cdot \mathbf{Trans}(0, l_2, 0) \\ \cdot \mathbf{Rot}(x, \theta_{m,3}) \cdot \mathbf{Trans}(0, l_3, 0) \cdot \mathbf{u}\end{aligned} \tag{3}$$

Where $\mathbf{Trans}(\cdot, \cdot, \cdot)$ and $\mathbf{Rot}(\cdot, \cdot)$ means translation and rotation matrix respectively. l_i represents the length of i-th leg part and $\theta_{m,i}$ is the rotation angle of joint i in m-th leg, symbol $\mathbf{u}$ stands for normal vector.

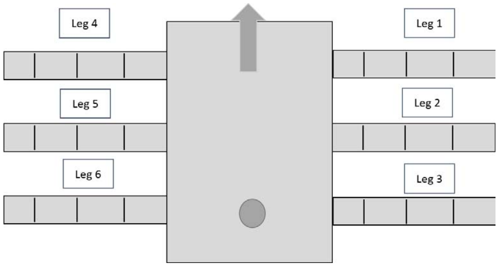

Fig. 2: Six-legged robot model

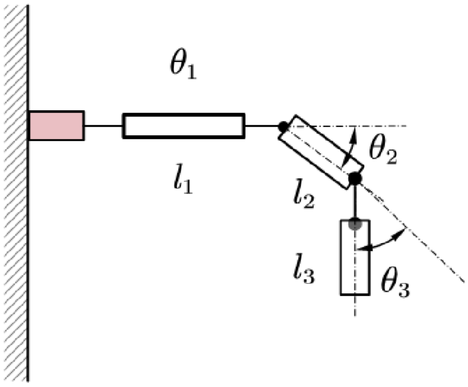

Fig. 3: Rotation angles of leg

III. Multi-objective Optimization Based Motion Generation

A. Encoding Scheme and Optimization Goals

In this research, we seek to solve a complex motion generation task on irregular terrains, and the walking behavior can be seemed as a series of postures of robot. The encoding scheme translates the motion from posture to leg and finally to joint angle level, and we use Fig 4 to depict the scheme briefly. Please note that this scheme is also used in the research [12]. Each posture is depended on six legs $L_i, (1 \leq i \leq 6)$ and each leg possesses three joint angle $\theta_j, (1 \leq j \leq 3)$, therefore the total number of variables of EDA is $18 * n$, where n is the number of postures.

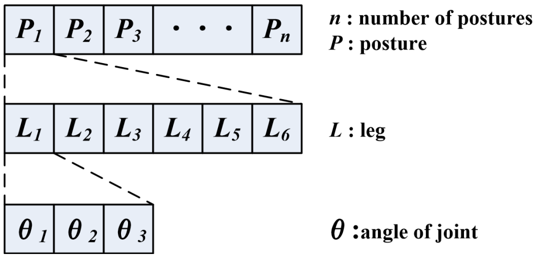
P_1 P_2 P_3 · · · P_n
n : number of postures
P : posture
L_1 L_2 L_3 L_4 L_5 L_6
L : leg
θ_1 θ_2 θ_3
θ :angle of joint

Fig. 4: Representation of individual

The motion generation problem aims to find a gait that maximizes some performance metric, which corresponds to the corresponding optimization goals. In our research, the optimization function is calculated according to the Energy Stability Margin (Equation 1) and the Froude number (Equation 2), which comes from :

$$\begin{aligned}f(\theta_{1,1}, \cdots, \theta_{6,3}) &= \min(\frac{1}{F_{r2}}, S_{ESM}) \\ &= \min(\frac{gh}{V}, \min\limits_{i}^{l_s}(mgh_i)) \\ &= \min(\frac{gh}{V}, \min\limits_{i}^{l_s}(mg(|R_i| \cos\psi(1 - \cos\theta))))\end{aligned} \tag{4}$$

In a word,the optimization goal is defined in the expectation that the robot move as far as possible and save energy. For a nontrivial multi-objective optimization problem, no single solution exists that simultaneously optimizes each objective, and the reason is that the objective functions are conflicting, so the solutions of the problems is a (possibly infinite) number of Pareto optimal solutions, and interested readers about Multi-objective Optimization Problem (MOP) and some basic concepts, such as Pareto dominate, Non-dominated set, Pareto-optimal Set (PoS), Pareto-optimal Front (POF) of the MOP can refer to [14].

B. Trend Prediction Model Based Multi-Objective Estimation of Distribution Algorithm

We proposed the TPM-EDA [17] to solve multi-objective optimization problem. In the TPM-EDA framework, a hyper-solution is a key concept and it is defined as follows. Please noted that in this research, a potential motion can be regarded as a special form of a hyper-solution.

Definition 1. *For an optimization problem with n objective functions, a hyper-solution is a tuple $P = \langle x, d \rangle$, where x is a n-dim vector representing the solution of the multiobjective* *optimization problem and* d *is a* n*-dim unit vector representing the direction of the hyper-solution.*

We use Fig. 5 to illustrate the steps of the TPM-EDA. This algorithm involves the following steps: At first, TPM-EDA generates some hyper-solutions (the black points in the Fig. 5 (a), and then it calculates the Non-dominated set [1] (the red points in the Fig. 5 (b). In the subsequent step, the TPM-EDA estimates the probability density of the distribution which corresponds to the Non-dominated set via a nonparametric kernel density estimation method, and then the proposed algorithm will select one hyper-solution by using important sampling (the Fig. 5 (c)), and the principle is that the hyper-solution with high sparseness value can get more chances to be selected and vice versa. After this step, the TPM-EDA will feed the selected hyper-solution into the proposed prediction model to generate some new hyper-solutions (the black points in the Fig. 5 (d)), and the next step will get updated Non-dominated set (Fig. 5 (e)) according to the hyper-solutions. The program will loop continuously until the termination condition is satisfied.

The basic idea of the generating offsprings of the TPM-EDA is to assign an importance weight, called sparseness, to every hyper-solution in the non-dominated set, and the weight can be considered as a probability value and it will change over time. The hyper-solutions with higher sparseness values will have a better chance of producing more offspring than the hyper-solutions with lower sparseness values.

Given a set of hyper-solutions $DP = \{p_1, \ldots, p_n\}$, where $p_i = \langle x_i, d_i \rangle, (1 \leq i \leq n)$ is a hyper-solution. The probability density of a hyper-solution p_k is calculated by kernel density estimation, that is:

$$Density\left(p_k\right) = \frac{1}{nh}\sum_{i=1}^{n}\kappa\left(\frac{x_k - x_i}{h}\right), \tag{5}$$

where $\kappa(\cdot)$ is the kernel function, and we used Gaussian kernel function in this research, which means that $\kappa(u) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}u^2}$. The parameter h adjusts the bandwidth of the probabilistic density curve. In other words, the probability density curve will be smooth when h is large, and conversely the curve is steep if the parameter is small.

The probability density describes the degree of congestion, which means if the value of the probability density is high, there are more hyper-solutions within this area; if the value is low, there are fewer hyper-solutions around. Sparseness of a hyper-solution is defined with the probability density.

Definition 2. *The sparseness of the hyper-solution* p_k *is defined by the reciprocal of the probability density:*

$$Sparseness\left(p_k\right) = \frac{1}{Density\left(p_k\right)}. \tag{6}$$

After choosing a hyper-solution, we need an effective means to produce more good offspring. A probabilistic inertial prediction model is presented in this section to achieve the goal. The input of this model is the individual selected by the method described in the above subsection and the output of this model is a set of hyper-solutions. Based on this model, a multi-objective optimization algorithm is proposed.

For the problem of how to decide the direction of the offspring hyper-solution, we propose a special distribution, called Probabilistic Inertial Distribution (PID), to solve it. Before describing our method in detail, we give the definition of the probability inertia distribution at first.

Definition 3 (Probabilistic Inertial Distribution, PID)**.** *Given a vector* $u \in \mathbb{R}^n$*, the probabilistic inertial distribution measures the probability of the angle* θ *between the sampled vector* v *and* u *:*

$$\mathrm{PID}_a^{u,v}\left(\theta\right) = \frac{e^{-a\theta}}{\int_0^{\pi} e^{-a\theta}d\theta},\ \theta \in \left[0, \pi\right], \tag{7}$$

where $a \in \mathbb{R}^{+}$*.* u *is called the inertial vector.*

The Procedure GenOffs to be detailed later describes how to generate offspring. After an angle θ is sampled from the inertial distribution, we will carry out the Procedure Randpick to pick out a vector from an infinite number of vectors whose angle to u is θ. And then a step length is sampled from a normal distribution. The Procedure 1 depicts a multi-objective optimaztion algorithm which is based on the Procedure GenOffs.

```
Procedure GenOffs($POS$, $a$, $N$)
  Input: $POS$: A set of hyper-solutions; $a$: A control
         parameter; $N$: The number of offspring;
  Output: A set of hyper-solutions;
  $samples \leftarrow \emptyset$;
  for $i = 1$ to $N$ do
      $p \leftarrow$ randome select from the non-dominated set
        $POS$;
      Sample $\theta$ from the distribution $PID_a$;
      $v$ = Rrandpick($p.direction, \theta$);
      Sample a length $\lambda$ from $Norm(0, 1/a)$ and
        normalized it to $[0, \infty]$;
      Add a new hyper-solution $p + \lambda v$ to $samples$;
  end
  return $samples$;
```

Please noted that the Δ function in the algorithm is defined as follows:

$$\Delta\left(P_1, P_2\right) = \frac{\sum\limits_{p \in P_1} \min\limits_{q \in P_2} \|p - q\|_2 + \sum\limits_{p \in P_2} \min\limits_{q \in P_1} \|p - q\|_2}{|P_1| + |P_2|}. \tag{8}$$

IV. Experimental results

A. Experimental environment

We perform experiments to test our method using Webots [18]. Webots is a professional environment to model, program and simulate mobile robots. The simulation environment is

[1] Non-dominated set is also called Pareto Set (PS), and the non-dominated front is denoted by PF in this paper.

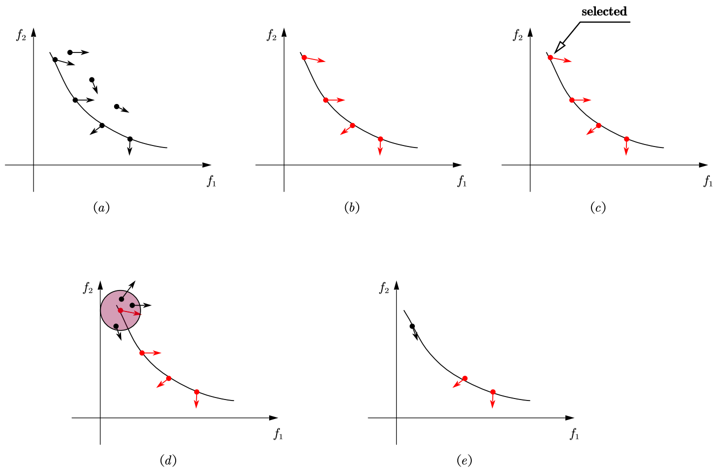

Fig. 5: The Five Steps of the TPM-EDA

```
Algorithm 1: TPM-EDA based Motion Generation($F(X)$, $Pop$, $h$, $r$, $N$)
Input: $F(X)$: The MOP is defined in Equation (4);
       $Pop$: the initialized population of ; $h$: the
       initialized bandwidth; $r$: the initialized
       attenuation coefficient; $N$ the number of
       samplings;
Output: $BestMotion$: A motion for the robot;
Get $POS^0$ and $POF^0$ by $Pop$;
$samples$ = GenOffs($Pop, 1/h, N$);
Get $POS^1$ and $POF^1$ by $samples$;
$POF$ = $POF^0$ ;
$POF'$ = $POF^1$ ;
$TempPOS$ = $POS^1$ ;
while $\Delta(POF, POF') > \varepsilon$ do
    /* The definition of $\Delta$ please
       refer to Equation (8).                    */
    $samples$ = GenOffs($TempPOS, 1/h, N$);
    $POF$ = $POF'$ ;
    Get $TempPOS$ and $POF'$ by $samples$;
    $h = h * r$;
end
$BestMotoin \leftarrow$ random selection an element from
  $TempPOS$;
return $BestMotion$;
```

shown in Fig 6. The task is difficult for robot because it has to keep the stability of the body, attempting to move forward as fast as possible as well as save energy. The parameters of robot model are shown in Table I. Table II illustrates the operation intervals and initial values of the joint angles.

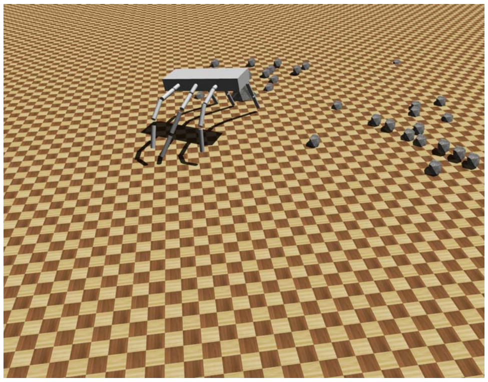

Fig. 6: Simulation environment

TABLE I: Parameters of the robot model

Parameter	Body	Leg
Mass(kg)	1.6	0.04
Length(m)	0.32	0.02
Width(m)	0.16	0.02
Height(m)	0.04	0.04

To verify the effectiveness of our approach, we set up different complex environments. More specifically, the number of obstacles in different environments is different, and this difference directly causes the difficulty of motion generation. We compare the movement speeds and energy consumptions which caused by different approaches and the experimental results are presented in Fig. 7 and Fig. 8. The first method is the traditional genetic algorithm, and we use the blue line to depict the results of the method and the other approach is the MGG approach which described in [12], [19].

TABLE II: Parameter of the joint angles

	1st joint	2nd joint	3rd joint
range	-45° $\leq \theta_1 \leq$ 45°	0°$\leq \theta_2 \leq$ 60°	0° $\leq \theta_3 \leq$ 60°
initial	0°	45°	45°

By looking at Fig. 7, we can find that the movement speed obtained by using the proposed method is almost the same as the other methods when the obstacle is small, however, with the increase of obstacles, our proposed method can improve the speed of the robot than other methods. Fig. 8 describes the energy consumed by the different methods. Similarly, it is not difficult to find that the method proposed in this paper can reduce the energy consumption of the robot, especially in the case of large number of obstacles.

V. Conclusion

In this article, we propose a motion generation method for multi-legged robot, and the basic idea of the method is that we convert the motion generation problem into a multi-objective optimization problem and then use an estimation of

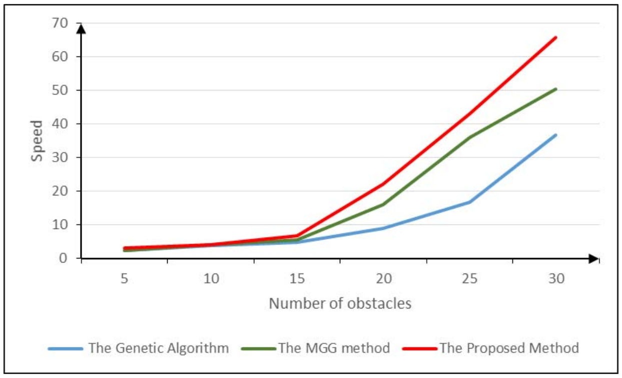

Fig. 7: Movement speeds of three types of robots when they across different rough terriers

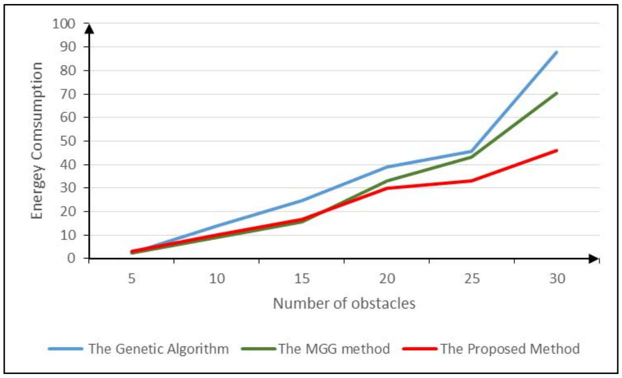

Fig. 8: Energy consumptions of three types of robots when they across different rough terriers

distribution algorithm to solve the optimization, such that we can obtain appropriate motions. Those motions not only make the robot move faster, but also make it more energy efficient. The experimental results show that the proposed algorithm is promising.

This research is a starting point, but it inspires us to think is it possible to reuse knowledge to improve the ability of the robot? Can we introduce a transfer learning technique [20] and if we can combine this kind of method with other approaches [21]–[23] to solve the motion generation problems?

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (No.61003014 and No.61673328), and the National Social Science Foundation (15BYY082).

REFERENCES

[1] Daoxiong Gong, Jie Yan, and Guoyu Zuo. A review of gait optimization based on evolutionary computation. *Applied Computational Intelligence and Soft Computing*, 2010, 2010.

[2] Sonia Chernova and Manuela Veloso. An evolutionary approach to gait learning for four-legged robots. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2562–2567. IEEE, 2004.

[3] Dilip Kumar Pratihar, Kalyanmoy Deb, and Amitabha Ghosh. Optimal path and gait generations simultaneously of a six-legged robot using a ga-fuzzy approach. *Robotics and Autonomous Systems*, 41(1):1–20, 2002.

[4] Jason Yosinski, Jeff Clune, Diana Hidalgo, Sarah Nguyen, Juan Cristobal Zagal, and Hod Lipson. Evolving robot gaits in hardware: the hyperneat generative encoding vs. parameter optimization. In *ECAL*, pages 890–897, 2011.

[5] Pandu Ranga Vundavilli and Dilip Kumar Pratihar. Soft computing-based gait planners for a dynamically balanced biped robot negotiating sloping surfaces. *Applied Soft Computing*, 9(1):191–208, 2009.

[6] Goswami Dip, Vadakkepat Prahlad, and Phung Duc Kien. Genetic algorithm-based optimal bipedal walking gait synthesis considering tradeoff between stability margin and speed. *Robotica*, 27(3):355–365, 2009.

[7] Milton Roberto Heinen and Fernando Santos Osório. Applying genetic algorithms to control gait of physically based simulated robots. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1823–1830. IEEE, 2006.

[8] János Botzheim, Noriko Takase, and Naoyuki Kubota. Cyclic motion generation for intelligent robot by evolutionary computation. In *Robotic Intelligence In Informationally Structured Space (RiiSS), 2013 IEEE Workshop on*, pages 13–19. IEEE, 2013.

[9] Sadayoshi Mikami, Hiroaki Tano, and Yukinori Kakazu. Acquiring adaptive gaits for many-legged robots by reinforcement learning. *Nippon Kikai Gakkai Ronbunshu, C Hen/Transactions of the Japan Society of Mechanical Engineers, Part C*, 60(580):4252–4259, 1994.

[10] Noriko Takase, János Botzheim, and Naoyuki Kubota. Evolving spiking neural network for robot locomotion generation. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 558–565. IEEE, 2015.

[11] Kazuyuki Ito and Fumitoshi Matsuno. Applying qdsega to the multi legged robot. *Transactions of the Japanese Society for Artificial Intelligence*, 17:363–372, 2002.

[12] Mutsumi Iwasa, Takenori Obo, and Naoyuki Kubota. Motion generation of multi-legged robot by using knowledge transfer in rough terrain. In *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*, pages 1–5. IEEE, 2016.

[13] Pedro Larranaga. A review on estimation of distribution algorithms. In *Estimation of distribution algorithms*, pages 57–100. Springer, 2002.

[14] Jürgen Branke, Kalyanmoy Deb, and Kaisa Miettinen. *Multiobjective optimization: Interactive and evolutionary approaches*, volume 5252. Springer Science & Business Media, 2008.

[15] Dominic Anthony Messuri. *Optimization of the locomotion of a legged vehicle with respect to maneuverability*. PhD thesis, The Ohio State University, 1985.

[16] Hiroshi Kimura, Isao Shimoyama, and Hirofumi Miura. Dynamics in the dynamic walk of a quadruped robot. *Advanced Robotics*, 4(3):283–301, 1989.

[17] Zhongqiang Huang and Min Jiang. Trend prediction model based multi-objective estimation of distribution algorithm. *Artificial Intelligence and Robotics Research*, 5(1), 2016.

[18] https://www.cyberbotics.com/webots.php.

[19] Hiroshi Satoh. Minimal generation gap model for gas considering both exploration and exploitation. In *Proceedings of 4th International Conference on Soft Computing, 1996*, pages 494–497, 1996.

[20] Min Jiang, Wenzhen Huang, Zhongqiang Huang, and Gary G Yen. Integration of global and local metrics for domain adaptation learning via dimensionality reduction. *IEEE Transactions on Cybernetics*, 47(1):38–51, 2017.

[21] Min Jiang, Changle Zhou, and Shuo Chen. Embodied concept formation and reasoning via neural-symbolic integration. *Neurocomputing*, 74(1):113–120, 2010.

[22] Min Jiang, Yulong Ding, Ben Goertzel, Zhongqiang Huang, Changle Zhou, and Fei Chao. Improving machine vision via incorporating expectation-maximization into deep spatio-temporal learning. In *IJCNN*, pages 1804–1811, 2014.

[23] Min Jiang, Yang Yu, Xiaoli Liu, Fan Zhang, and Qingyang Hong. Fuzzy neural network based dynamic path planning. In *Machine Learning and Cybernetics (ICMLC), 2012 International Conference on*, volume 1, pages 326–330. IEEE, 2012.