

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Evolutionary Multi-Objective Optimization for Biped Walking of Humanoid Robot

Toshihiko Yanase and Hitoshi Iba
The University of Tokyo
Japan

1. Introduction

The recent remarkable progress of robotics research makes advanced skills for robots to solve complex tasks. The divide-and-conquer approach is an intuitive and efficient method when we encounter complex problems. Being a divide-and-conquer approach, the multi-layered system decomposes the problem into a set of levels and each level implements a single task-achieving behaviour. Many researchers employ this approach for robot control system, dividing a complex behaviour into several simple behaviours. For example, Lie et al. developed the Evolutionary Subsumption Architecture, which enables for heterogeneous robots to acquire the cooperative object transferring task (Liu & Iba, 2003).

The autonomous locomotion of humanoid robots consists of following modules: global path planning using given geometrical information, local path planning based on the observation of environment, footstep planning, and whole-body motion generator. Since these modules mainly exchange the information with their neighbours, we can observe that they are hierarchically arranged from the aspect of communication. The parameter settings among these modules are necessary to adapt the system to the targeting environment. The problem involves a number of conflicting objectives such as stability of the robot motion and speed of locomotion.

In this paper, we present a parameter tuning method for multi-layered robot control system by means of Evolutionary Multi-Objective Optimization (EMO). We explore the set of parameters for modules to adapt various kinds of environments. Switching these parameter sets enables us to operate the robots effectively. We developed three modules as the experiment environment: walking pattern generator, footstep planner and path planner. In the experiment, we focused on the footstep planner shown in Fig. 1, which realizes collision-free walking. The parameter setting was manually adjusted in previous researches (Kuffner et al., 2003; Chestnutt et al., 2005). We discuss the conflicting objectives for the optimization footstep planner, and introduce a parameter setting method using EMO. Then we propose a simple rule to use parameter sets obtained by EMO to adapt the footstep planner to both crowded and sparse fields.

The rest of the paper is organized as follows: Section 2 describes our robot control system, Section 3 shows an experiment of the parameter setting of the footstep planner, and Section 4 shows an application using the parameter setting obtained from Section 3 and a

comparison with a conventional approach. Section 5 provides the discussion of these experiments, and finally Section 6 concludes with summary and future works.

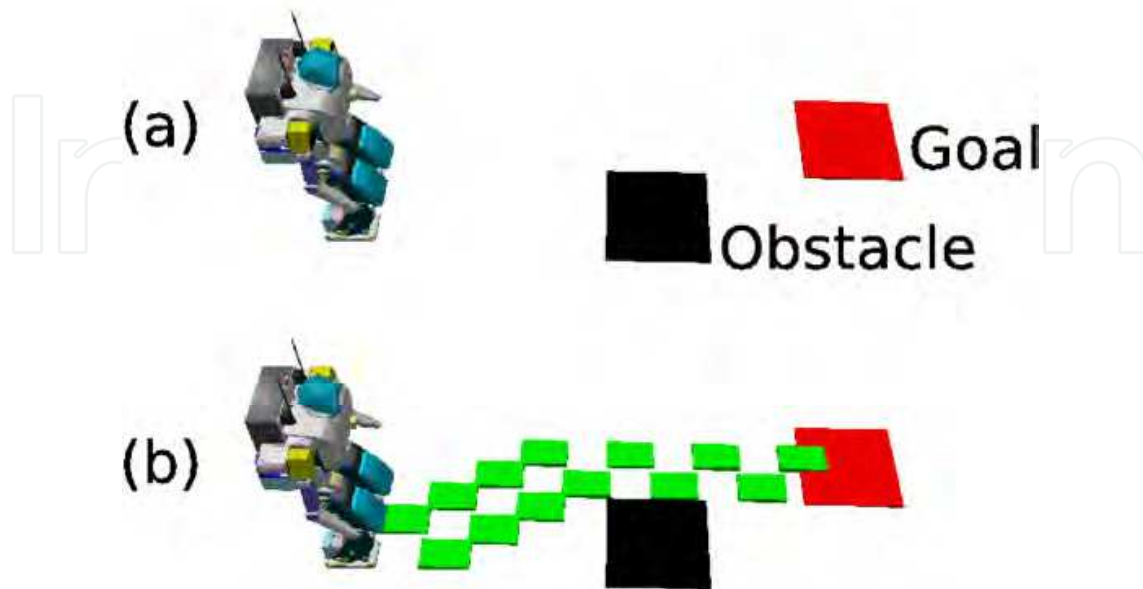


Fig. 1. Example of Footstep Planning for the humanoid HOAP-1

2. Multi-layered Robot Control System

2.1 System Overview

We implemented a locomotion control system which is able to find an optimal sequence of footsteps from a start location to a goal position without colliding with obstacles. The system consists of three modules, path planner, footstep planner and whole-body motion generator. The path planner generates the geometrical path for the given environmental map. Then footstep planner calculates a sequence of footsteps based on the path information. The whole-body motion generator calculates the trajectories of joint angles which realize the given sequence of footstep without losing the stability of the robot motion. By using these modules cooperatively, the locomotion planning is executed as follows:

1. Plan the entire path from the start position to the goal position by using RRT. (Fig. 2 (a))
2. Place the sub-goals for footstep planner based on the length of path at regular intervals. (Fig. 2 (b))
3. Plan the sequence of footsteps to the sub-goals. (Fig. 2 (c))
4. Repeat Step 3 until the robot reaches the goal position.

In our research, we have used the HOAP-1 (Humanoid for Open Architecture Platform) robot manufactured by Fujitsu Automation. We adjusted the constraints of walking, such as the limitation of landing position and joint angles for HOAP-1. Motions were controlled by specifying the joint angles of the 20 joints of the entire body every 0.002 seconds. The characteristics of the HOAP-1 are noted below: (1) Height: 483 [mm], Weight: 5.9 [kg]. (2) The internal interface between the hardware and software is available for public use. (3) For

movable parts, each leg had six degrees of freedom and each arm had four degrees of freedom, for a total of 20 degrees of freedom on the left and right sides.



Fig. 2. (a) Path from the initial position to the goal position planned by RRT algorithm. (b) Sub-goals placed on the path. (c) Footstep sequence generated by A* Search algorithm.

2.2 Path Planning

We implemented the path planning module based on the Rapidly-exploring Random Tree (RRT) algorithm (Kuffner et al., 2000). RRT algorithm is a probabilistic motion planner which has found widespread use in the field of robot motion planning (Kamio & Iba, 2006; Zucker et al., 2007). Because RRT offers almost no hypothesis in terms of the problem, we could easily apply it to various problems. In this study, the state of the robot was expressed using 2D information (x coordinate, y coordinate), and we did not deal in another kind of information such as the rotation angle around z axis and the velocity of the robot to simplify the problem. We have to examine the collision of robots and obstacles when we use the path planning. In this experiment, the robot was approximated by the circle whose radius is 20 [cm] in order to detect the conflict between obstacles and the robot.

Before generating a sequence of footsteps which satisfies the planned path, we place the sub-goal for footstep planner. The sub-goal is decided by the given path length because the computational cost for the footstep planning increases with the number of footsteps.

2.3 Footstep Planning

A footstep planner returns an optimal sequence of footstep locations according to the plausible sets of footstep locations defined as shown in Fig. 3. In this research, we employed a footstep planner based on the A* Search Algorithm (Kuffner et al., 2003; Chestnutt et al., 2005). This footstep planner is fast enough to be used for the real environment. This is because the planner discretize the footstep location and explores footstep sequences in order of the estimated cost. Next, we explain the setup of A* Search in this research.

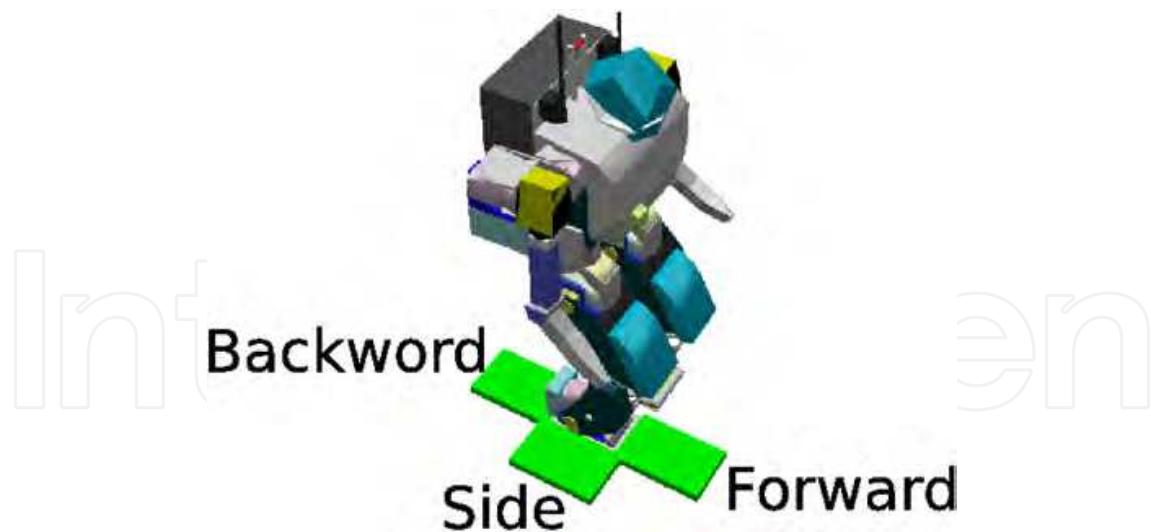


Fig. 3. Example of footsteps of the humanoid HOAP-1

2.3.1 Finite Set of Actions

The A* Search Algorithm uses a finite set of actions for exploration. In case of footstep planning, we define an action as the candidate footstep at the next time step, hence the candidate footstep should be discretized in advance. Fig. 3 shows an example of a set of footsteps which is discretized into four candidates. The candidate footsteps are placed symmetrically when the robot switch the supporting legs.

2.3.2 Environmental Map

We used two kinds of maps as the experimental environment. At first, we employed a grid map in the experiment of Section 3. Each grid is uniformly-shaped and has information about the existence of obstacles. The other one is the vector map. Obstacles and walls are represented by rectangles in the vector map which is used in Section 4. The collision between footprints and obstacles is detected in both these maps.

2.3.3 Estimation of Cost

The A* Search Algorithm calculates the cost of the entire path by using the cost function and the heuristic estimation. The cost function is the length of footstep sequence from the initial position to the current position. The heuristic estimation calculates the minimum length of the footstep sequence from the current position to the goal position by dividing the distance by the maximum step length.

2.3.4 Termination Criteria

The search is terminated when the footstep sequence reaches the goal position. The search can fail in one of two ways:

- The number of A* search tries exceeded the maximum allowable. In this experiment we limited the maximum number to 3000.
- No more valid successor nodes can be generated. In this case, no collision-free footstep sequence exists using the given discrete set of footstep locations.

2.4 Whole-body Motion Generator

Up to now, many biped robots are successfully developed with the Zero Moment Point (ZMP) concept (Zhu et al., 2004; Nishiwaki et al., 2002; Nishiwaki & Kagami, 2006). In this experiment, one of the ZMP based method, a Fast Generation Method of Motion Pattern that Follows Desired ZMP (Nishiwaki et al., 2002) proposed by Nishiwaki et al. was used as a whole-body motion generator. The features of this method are as follows: (1) realizing the expected feet position accurately and (2) fast enough to be used as a real-time module. Since real robots have many performance constraints, the gap between the simulated and real robot becomes a problem. Following performance constraints are taken into consideration in this paper; (1) Existence of the solution for inverse-kinematics of legs, (2) Limitation of joint angle range, (3) Collision between links. We examined the locations of feasible states by using the simulated model of HOAP-1, and stored in the regular grid. The footstep planner uses the regular grid as a reference to detect infeasible footstep locations.

2.5 Problems of Current Biped Walking Control System

The A* Search Algorithm is used for exploring a graph as shown in Fig. 4, which represents the footstep transitions. The node S means the initial state, and each node represents the footstep locations. The red node means the goal state. The gray node means an infeasible footstep which can not be applied at the time step. The number in the footstep location shows the kind of footstep.

Fig. 4 (a) shows a search with the footstep sequence with footstep set {1, 2, 3}, and failed to reach to the goal state. There are two kinds of countermeasures. One is to replace the set of footsteps with {1, 4, 5} as shown in (b), and the other one is to add the footstep {4, 5} as shown in (c). As can be seen from Fig. 4 (b) and (c), both methods successfully reach to the goal state, however, (c) requires more nodes than (b). In the Fig. 4 (a) and (b), three kinds of candidate footsteps are used for the footstep planning, so that the possible search space is a full 3-ary tree. On the other hand, the possible search space of Fig. 4 (c) is full 5-ary tree. The depth of the graph means number of footsteps. The spread of graph changes depending on the number of candidate footsteps. The maximum number of node in the search tree is represented as follow:

$$N_{\max} = \frac{1 - k^n}{1 - k}, \quad (1)$$

where k is the number of candidate footsteps and n is the length of the footstep sequence. The search space grows significantly as the increase of candidates. For example, in case of $n = 10$ and $k = 3$, the maximum number of node is 29524. In case of $n = 10$ and $k = 5$, the maximum number of node is 2441406. We have to reduce the number of candidate footstep in order to calculate long footstep sequences with the same amount of search time. In this research, we propose a switching method of footstep sets in order to curb the growth of search space.

We can also observe the side-effect of discretized footstep locations. If we employ only short footsteps, the robot is able to pass through narrow paths. However, such kind of set of footsteps usually requires much computational resources for the footstep planner because short step increases the total steps. On the other hand, the set of landing positions which has only long footsteps reduces the total steps and it increases the feasible path width. In this

research, we focused on two requirements, to reduce the computational resources and to minimize the feasible path width.

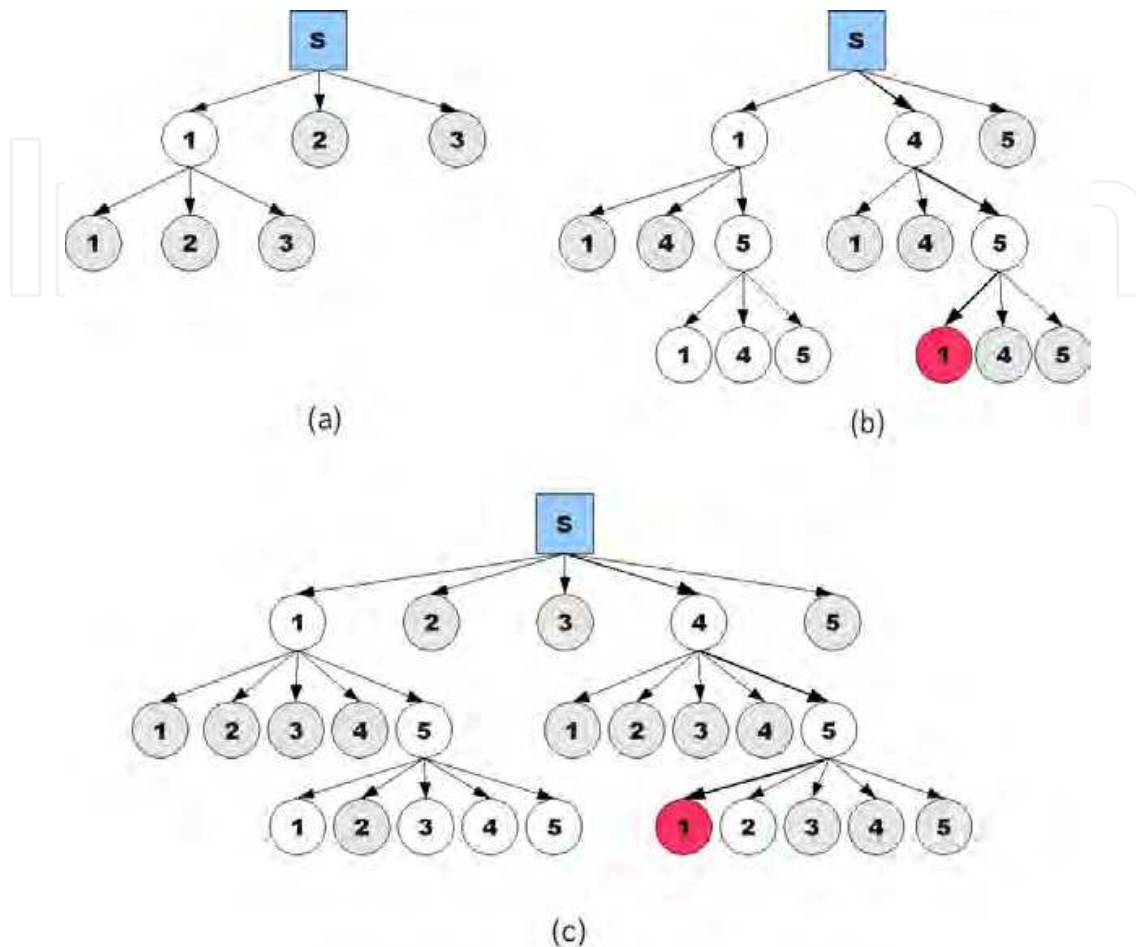


Fig. 4. Search space of A* Search Algorithm. Each node represents a state, which is a footstep location. Gray nodes mean infeasible states. The red nodes mean goal states.

3. Multi-Objective Optimization of Footstep Planner

3.1 Evolutionary Multi-Objective Optimization

The multi-objective optimization problem has multiple criteria to be minimized or maximized. To solve this problem we need to explore a set of solutions called Pareto optima. Evolutionary Multi-Objective Optimization which is one of the methods to solve the multi-objective optimization problem is rapidly developed by many researchers. Representative methods are as follows: Multi-Objective Genetic Algorithm (MOGA) (Fonseca & Fleming, 1993), Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002), and Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler et al., 2001). These methods are widely used in various kinds of research fields (Coello Coello, C. & Lamont, 2004).

3.2 Experimental Setup

We used NSGA-II which is an elitism EMO algorithm for optimization of the discrete set of landing locations. NSGA-II employs the mechanism to penalize the crowded solutions in

order to keep the diversity of solutions. NSGA-II was widely applied for various Real-World Applications (Deb et al., 2002). In this experiment, a chromosome was composed of two real variables. Each variable corresponds to the length of the footstep, forward-backward and sideward.

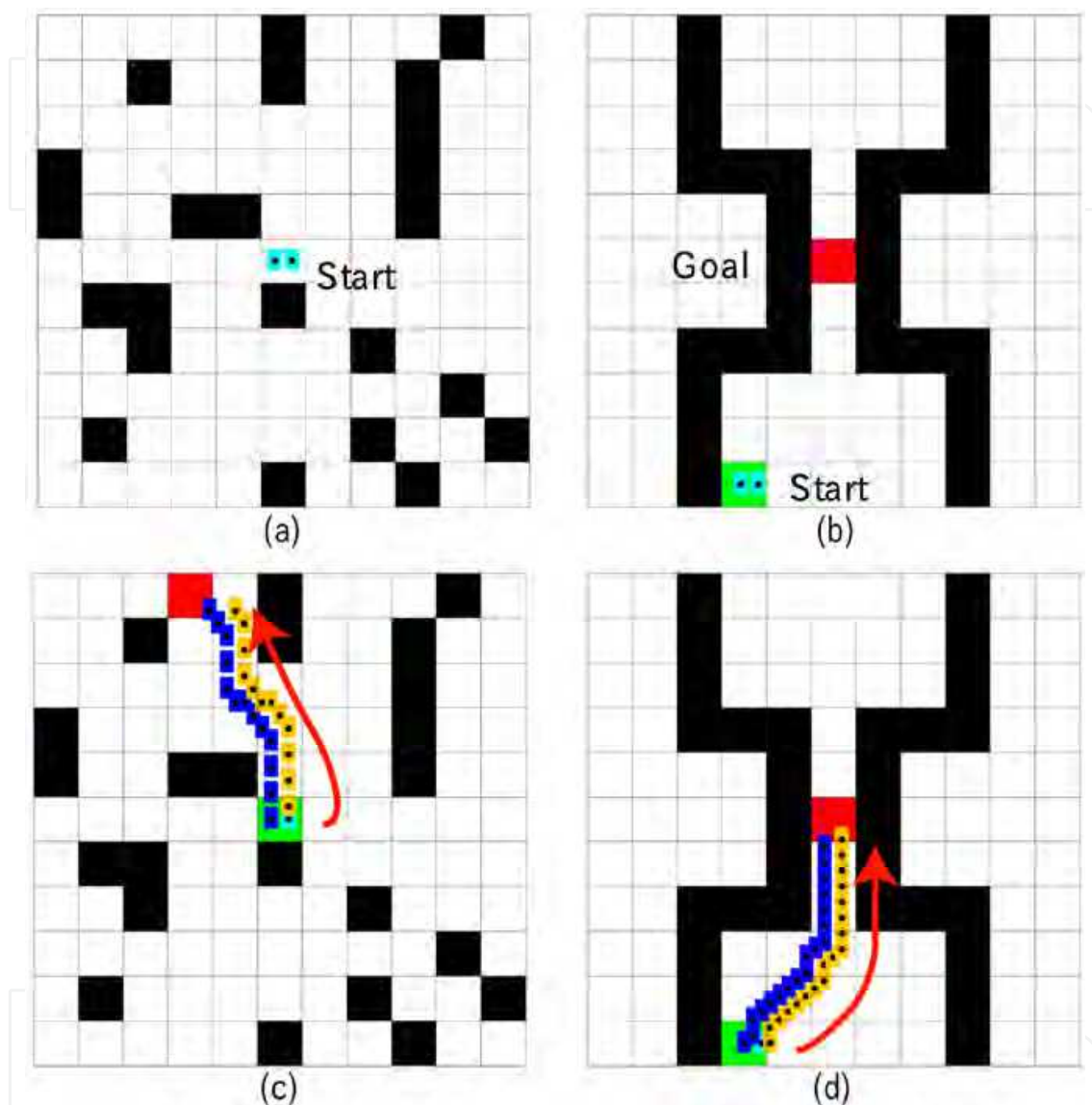


Fig. 5. Environmental maps for optimization. (a) Environment A to evaluate average numbers of A* search tries (b) Environment B to evaluate minimum feasible width of path. (c) and (d) show typical results of footstep planning.

As we mentioned in Section 2, two conflicting objectives exist. The criteria of this optimization were as follows:

- Average numbers of A* search tries
- Minimum feasible width of path

In order to evaluate these two objective functions, we created two environments: Environment A (Fig. 5 left, 220cm × 220cm), Environment B (Fig. 5 right, 220cm × 220cm). With Environment A, we measured average numbers of A* search tries to evaluate the

computational cost. We placed the obstacles in Environment A randomly. The start position was settled in the center of the map and the goals were settled in each free space. The footstep planning was carried out from left-top to right-bottom respectively. After evaluating all the free space, we averaged the numbers of A* search tries. With Environment B, we evaluated the second objective. The task is to reach the goal located in the middle of the narrow path. We simply explored the minimum width by using hill-climbing method.

The parameters of NSGA-II are listed in Table 1. The population number was set to 100. The number of generation was set to 70. We employed real-valued mutation based on the uniform distribution as the mutation operator, and the Unimodal Normal Distribution Crossover (UNDX) (Ono et al., 2003) as the crossover operator. The Two parameters of UNDX were determined according to the recommended values in the literature (Ono et al., 2003 ; Ono & Kobayashi, 1997). The crossover rate was 0.9, and mutation rate was 0.01. We assumed that all landing positions in the candidates can transfer to each other because we set the step cycle slow enough to satisfy the constraints.

Parameter Name		Value
Population Size		100
Maximum Generation		70
Crossover	Rate	0.9
	Method	UNDX
	δ^2_{ξ}	0.25
	δ^2_{η}	0.06125
Mutation	Rate	0.01
	Method	uniform mutation

Table 1. Parameter setup of NSGA-II

3.3 Results

At 70th generation we observed that most of the individuals became non-dominated solutions. We also applied 10000 times Random Search as a comparison. The nondominated solutions acquired by NSGA-II and Random Search in the typical run were plotted in Fig. 6. The vertical axis means the minimum feasible width of path [mm], and the horizontal axis means the average number of A* search tries. From the Fig. 6, we can observe the trade-off relationship between the two objectives. Both of the method successfully obtained the Pareto solutions, however we can observed the diversity of NSGA-II population is superior to the one of Random Search population.

We used the Ratio of Non-dominated Individuals (RNI) to compare the performance of the two populations. RNI is used for the comparison of two sets of non-dominated individuals S_1 and S_2 . At first, the united set of non-dominated individuals S_u is created by combining S_1 and S_2 . Then, RNI is calculated by the ratio of S_1 or S_2 among S_u . We say that one set of solutions is closer to the true Pareto-solutions if its RNI is greater than that of the other one. The RNI of NSGA-II was 63% and the RNI of Random Search was 37%. NSGA-II saved the number of evaluations by 30%, and successfully obtained better solutions.

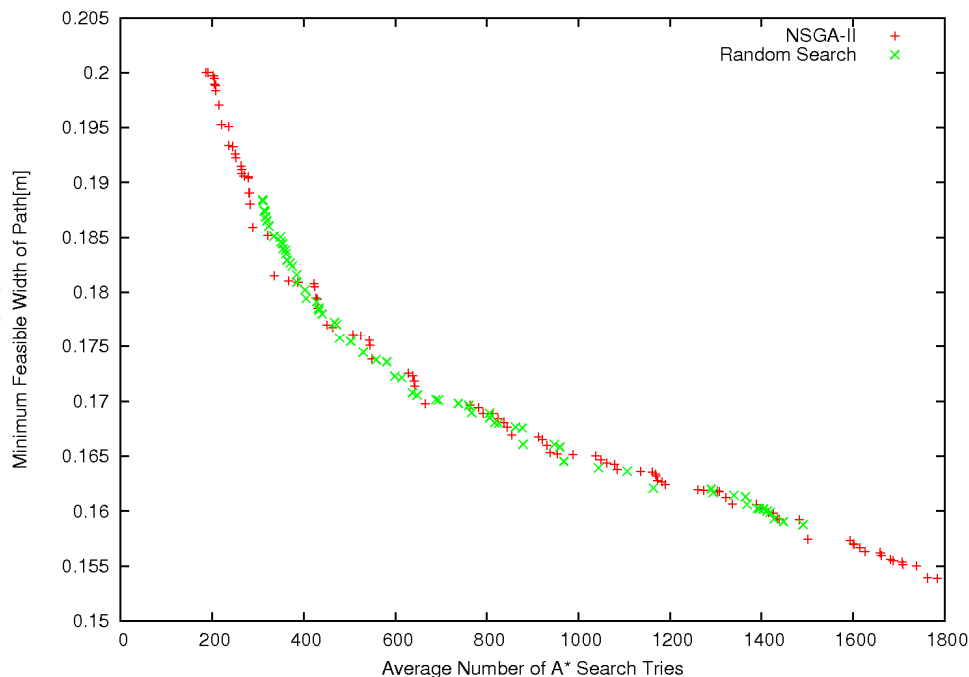


Fig. 6. Pareto-front of NSGA-II and Random Search

4. Switching the Parameter of Footstep Planner

4.1 Experimental Setup

In order to examine the performance of the acquired sets of landing position, we conducted another experiment. The validation environments are shown in Fig. 7. There are four rooms ($2\text{[m]} \times 2\text{[m]}$) connected by the corridor ($4\text{[m]} \times 0.5\text{[m]}$). The rectangle obstacles ($0.15\text{[m]} \times 0.04\text{[m]}$) were randomly placed in the environments. We fixed the start position at the right-bottom room and the goal position at the left-top room. The red circles represent the positions of the goals for the successful trials or the sub-goal where the planner failed. After generating the entire path from the initial position to the goal position by using the path planner, we added small obstacles ($0.04\text{[m]} \times 0.04\text{[m]}$). These small obstacles ignored by the path planner should be avoided by the footstep planner.

4.2 Switching Rule

In order to reduce the number of footsteps, we switched the sets of landing positions. We used two extreme sets of landing positions, which located at both edges of the Pareto front, the set with the smallest average number of A* search tries (Set A) and the set with the smallest minimum feasible width of path (Set B). The Set A was used to reduce the total footsteps, and the Set B redeemed the failures of the plans with Set A. In this experiment, we employed a simple rule to switch the sets of landing positions as follows:

1. Execute the footstep planning by using the Set A.
2. If the planner fails to create the footsteps in step 1, apply the Set B. Otherwise, go to step 1.
3. If the sequence of footsteps is successfully calculated, go to step 1. Otherwise, the plan fails.

4.3 Results

We compared the proposed method to the conventional method which explored the footsteps with a single fixed footstep set, the Set A or the Set B. The performance was evaluated by the total number of footsteps. Fig. 7 shows the experimental environments and a typical result of the path planning and the footstep planning. (a)-(c) in Fig. 7 shows the Environment C. (d)-(e) in Fig. 7 shows the Environment D. The difference between Environment C and D is the placement of the small obstacles. (a)(d) shows the footsteps planned with Set A. (b)(e) shows the footsteps planned with Set B. (c)(f) shows the footsteps planned by the proposed method. The proposed method generated the short sequences of footsteps and the sequence which was able for the robot to pass through the narrow space. On the other hand, the planner with Set A failed at the middle of the map where the small obstacles and the wall formed a narrow path as shown in (a). The planner with Set B successfully passed through the narrow path, however it generated much longer sequences of footstep than the proposed method. The total number of footsteps, averaged over 5 runs are presented in Table 2. The total number of footsteps of the proposed method was about 72.2 % less than that of planner with Set B.

	Planner with Set A	Planner with Set B	Proposed method
Environment C	-	1548	454
Environment D	398	1509	396

Table 2. Total number of footsteps

5. Discussion

5.1 Multi-Objective Optimization of Footstep Planner

As we discussed in section 2, we can observe the trade-off relationship between two performance measures of footstep planner, the computational cost and minimum feasible width of path. In the previous section, we successfully acquired the discrete sets of landing positions for the footstep planner. By using NSGA-II, we saved the number of evaluations by 30%, and successfully obtained better solutions compared to the Random Search.

The trade-off relationship in this experiment is caused by the grain size of discretization of search space, hence this kind of multi-objective problem can be also observed in other problems when we handle the continuous problems by the discrete algorithm.

5.2 Parameter Switching of Footstep Planner

We also introduced a simple rule to switch the sets of landing positions. Using the proposed method, we obtained the 72.2 % shorter sequences of footsteps than that generated by the planner with Set B. We also generated the sequence which was able for the robot to pass through the narrow space where the planner with Set A failed.

Our proposed switching method with two sets of candidates worked efficiently in the environment where only one narrow path exists in the sparse field. In case of the path whose width gradually changes, however, switching more sets of candidates will be more effective than the current method. We can simply change the switching rules and choose appropriate sets of candidate from many of them because we have explored Pareto-optimal solutions in advance.

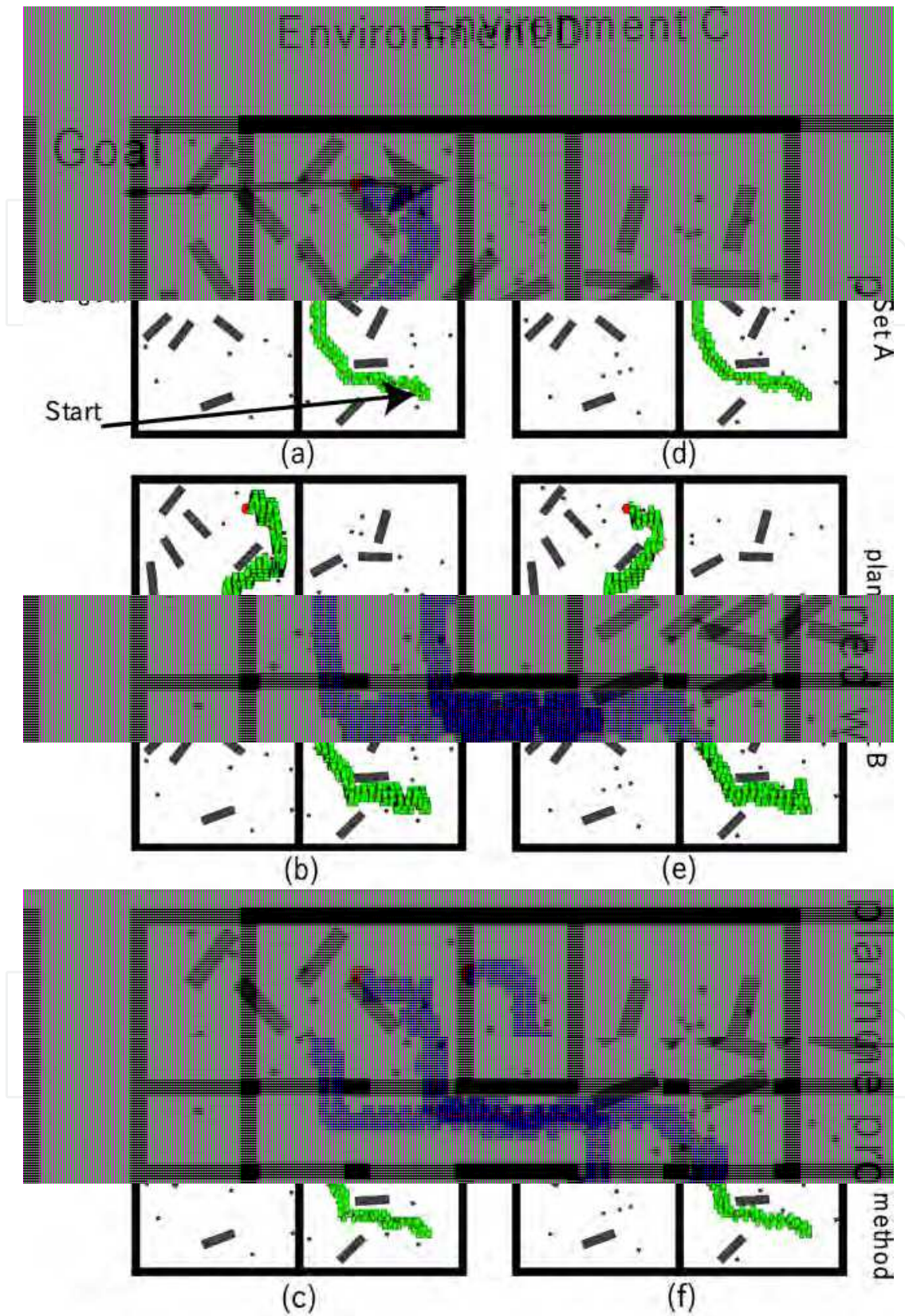


Fig. 7. Environment for validation: (a)(b)(c) show Environment C and typical results. (d)(e)(f) show Environment D and typical results. (a) failed to plan the footstep sequence at the narrow part of the path. (c) changed the set of footsteps at the narrow part of the path.

6. Conclusion and Future Works

We have presented a parameter setting method for a biped walking controller. At first, we introduced the robot control system and discussed the trade-off relationship of parameter settings for the footstep planning module. Next, we applied NSGA-II to acquire a set of parameters for the footstep planner. Then we proposed a simple rule to switch the sets of parameters to obtain the short sequence of footsteps. We also confirmed the proposed method generated the footsteps which was able for the robot to pass through the narrow space.

Our current work focuses on the translation of landing position without taking into consideration the rotation. We plan to optimize both parameters simultaneously to improve the performances of the footstep planner. Also, we intend to conduct experiments applying this system to the humanoid robot in a real environment.

7. References

- Chestnutt, J.; Lau, M.; Cheung, G.; Kuffner, J.; Hodgins, J. & Takeo Kanade. (2005). Footstep Planning for the Honda ASIMO Humanoid, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 629-634, April, 2005.
- Coello Coello, C. & Lamont G. (2004). An Introduction to Multi-Objective Evolutionary Algorithms and Their Applications, In : *Application of Multi-Objective Evolutionary Algorithms, Advances in Natural Computation*, Vol . 1, pp. 1-28, 2004.
- Deb, K.; Pratab, A.; Agrawal, S. & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-II, *IEEE Transaction on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197, 2002.
- Fonseca, C. M. & Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp 416-423, Morgan Kaufmann, 1993
- Kuffner, J. Jr. & LaVall, S. M. (2000). RRT-connect: An efficient approach to single-query path planning, *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '00)*, 2000.
- Kamio, S. & Iba, H. (2006). Cooperative object transport with humanoid robots using rrt path planning and re-planning, *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS2006)*, 2006.
- Kuffner, J; Kagami, S.; Nishiwaki, K.; Inaba, M. & Inoue. H. (2003). Online footstep planning for humanoid robots. *Proceedings of IEEE International Conference on Robotics and Automation 2003*, pp. 932-937, 2003.
- Liu, H. & Iba, H. (2003). Multi-agent learning of heterogeneous robots by evolutionary subsumption. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2003)*, pp. 1715-1728, 2003.
- Nishiwaki, K.; Kagami, S.; Kuniyoshi, Y.; Inaba, M. & Inoue, H. (2002). Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired zmp. *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pp. 2684-2689, 2002.
- Nishiwaki, K. & Kagami, S. (2006). High frequency walking pattern generation based on preview control of zmp. *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, pp. 2667-2672, 2006.

- Ono, I. & Kobayashi, S. (1997). A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover. *Proceedings of 7th Intl. Conference on Genetic Algorithm*, pp. 246-253, 1997.
- Ono, I.; Kita, H. & Kobayashi, S. (2003). A Real-coded Genetic Algorithm using the Unimodal Normal Distribution Crossover. In: *Natural Computing Series archive Advances in Evolutionary Computing: theory and applications*, pp. 213-237, 2003.
- Zhu, C.; Tomizawa, Y.; Luo, X. & Kawamura, A. (2004). Biped walking with variable zmp, frictional constraint, and inverted pendulum model. *Proceedings of IEEE International Conference on Robotics and Biomimetics 2004. (ROBIO 2004)*, pp. 425-430, 2004.
- Zitzler, E.; Laumanns, M. & Thiele, L. (2001). Spea2: Improving the performance of the strengthpareto evolutionary algorithm. *Technical Report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich*, 2001.
- Zucker, M.; Kuffner, J. & Branicky, M. (2007). Multipartite rrts for rapid replanning in dynamic environments. *Proceedings of IEEE Int. Conference of Robotics and Automation*, 2007.

IntechOpen

IntechOpen

IntechOpen



Climbing and Walking Robots

Edited by Behnam Miripour

ISBN 978-953-307-030-8

Hard cover, 508 pages

Publisher InTech

Published online 01, March, 2010

Published in print edition March, 2010

Nowadays robotics is one of the most dynamic fields of scientific researches. The shift of robotics researches from manufacturing to services applications is clear. During the last decades interest in studying climbing and walking robots has been increased. This increasing interest has been in many areas that most important ones of them are: mechanics, electronics, medical engineering, cybernetics, controls, and computers. Today's climbing and walking robots are a combination of manipulative, perceptive, communicative, and cognitive abilities and they are capable of performing many tasks in industrial and non- industrial environments. Surveillance, planetary exploration, emergence rescue operations, reconnaissance, petrochemical applications, construction, entertainment, personal services, intervention in severe environments, transportation, medical and etc are some applications from a very diverse application fields of climbing and walking robots. By great progress in this area of robotics it is anticipated that next generation climbing and walking robots will enhance lives and will change the way the human works, thinks and makes decisions. This book presents the state of the art achievements, recent developments, applications and future challenges of climbing and walking robots. These are presented in 24 chapters by authors throughout the world. The book serves as a reference especially for the researchers who are interested in mobile robots. It also is useful for industrial engineers and graduate students in advanced study.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Toshihiko Yanase and Hitoshi Iba (2010). Evolutionary Multi-Objective Optimization for Biped Walking of Humanoid Robot, Climbing and Walking Robots, Behnam Miripour (Ed.), ISBN: 978-953-307-030-8, InTech, Available from: <http://www.intechopen.com/books/climbing-and-walking-robots/evolutionary-multi-objective-optimization-for-biped-walking-of-humanoid-robot>

INTECH
open science | open minds

InTech Europe

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

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

www.intechopen.com

www.intechopen.com

IntechOpen

IntechOpen

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen