Distributed Human-based Genetic Algorithm Utilizing A Mobile Ad Hoc Network

Ryosuke Hasebe*, Kei Ohnishi*, and Mario Köppen*
*Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology
680-4 Kawazu, Iizuka-shi, Fukuoka 820-8502, Japan
Email: {hasebe@infonet.cse, ohnishi@cse, mario@ndrc}.kyutech.ac.jp

***Abstract*—A human-based genetic algorithm (HBGA) is one type of genetic algorithms, in which humans conduct all genetic operators such as selection, crossover, and mutation in a way such that they select others' solution candidates (selection) and create new candidate solutions influenced by the selected ones (crossover and mutation). HBGA needs a way for people to share their candidate solutions. One way is to manage candidate solutions in a centralized manner as a message board of a web forum, and actually such a HBGA has been implemented. However, how to implement HBGA in a distributed manner has not been well-studied so far. This paper presents a method for sharing candidate solutions among humans in HBGA running on a mobile ad hoc network (MANET), which is a distributed system, and shows simulation results to demonstrate the basic usefulness of the proposed method.**

***Index Terms*—genetic algorithm, human, ad hoc network, human interest.**

I. Introduction

Genetic Algorithms (GA) have been introduced as a large number of optimization methods and techniques that imitate biological evolution. The main processing steps of a GA fall into two categories that can be called "selection" (to mimic natural selection) and "operating" (to mimic mutation and crossover). But these processings can also be seen as the actions of agents, thus referring to the whole GA as a multi-agent system [1].

One may further distinguish between "Computer" agents (C) and "Human" agents (H) to do the processing. If both processing categories are performed by C-agents, the algorithm is a standard GA, or perhaps a parallel GA [2]. If the selection is performed by H-agents, the algorithm becomes an Interactive GA (IGA) [3] and if H-agents perform selection as well as operating, it becomes Human Based GA (HBGA) [1]. In this manner, the application area of GA has been widely increased.

In addition to performing selection, crossover and mutation operators by single agents, for processing the multi-agent system as a whole it needs scheduling agents as well. That is, one also has to consider the control of the interactions between the agents. For scheduling the processing of selection and operating in the standard GA or parallel GA, where there are only C-agents, the scheduling agent has to be a "Computer" modality as well. As for HBGA, an agent controls the timing of the human interactions as well, but here it may happen in various ways. For example, the two complementary methods listed in the following are possible [1]. One method is to establish a centralized management as only way to allow for the interaction between the agents in order to produce candidate solutions. For example, H-agents can communicate through a message board of a web forum [4]. In this method, while H-agents perform the calculations of the algorithm, the means of interaction is a C-agent.

The other method is to permit the agents to directly interact with the information and communication equipment, resulting into a distributed management of the problem solving H-agents. As a consequence, also selection, crossover and mutation are performed in a distributed manner. In this case, timing of agent interaction becomes humanized.

The specific method of organizing agents in HBGA has to provide both: the (physical) locations for selection and operating processing, and the management of potential solutions. The methods considered so far were depending on the communication technologies that were available at the time of the HBGA proposal. Recently, mobile ad hoc networks (MANET) [5] that allow to make wireless connections between moving wireless communication terminals have attracted attention. Obviously, they could be also considered to organize the agent interaction of a HBGA. As a network that is autonomously formed by agents, it would be a realization of the second method listed above.

MANETs can be formed when wide-area communication means are not available, like in the case of a disaster, and people have to resort to direct radio communication between their mobiles or smartphones [6][7]. Then, a real world example for using HBGA can be to find the best solution to escape from current danger area, taking knowledge and information from agents outside of the current area into account.

In this paper, the opportunities of using MANETs for HBGA are discussed. Especially a method for sharing the candidate solution between agents has to be devised. In order to confirm validity of the approach, results of a simulation in a simplified model are presented. In Section II related research is presented. In Section III, the organization of H-agents in a HBGA utilizing MANET to distribute candidate solutions is presented. Section IV then provides simulation results to demonstrate effectiveness of the proposed method. The paper concludes with a summary in Section V.

II. Related Work

HBGA[1] was first applied to a problem described in a natural language that requires us to find solutions that are also described in a natural language [1][8]. This first application of HBGA was build as a centralized system that uses a web site as a place of interactions among H-agents. However, as mentioned in [9], a range of applications of HBGA is not limited to problems descried in a natural language that humans easily understand. In [9], the effectiveness of H-agents who conduct crossover and mutation operators has been shown by the performance comparison of IGA and HBGA for a problem that is not described in a natural language.

In addition, the framework of GA has been utilized for problem solving in human organizations and for enhancing human creativity [8][10]. In [8], components and procedures in human organizations are regarded as basic GA components of genes, individuals, and a population, as well as basic GA procedures of selection, crossover, and mutation. In [10], data mining techniques are first applied to discussion logs for some problem solving on a Web message board in on-line manner. Next, building blocks (words) useful for the problem solving are extracted. Finally, the building blocks are fed back to the participants. The iterations of this procedure is regarded as a process through which GA produces good quality of solutions by combining the building blocks.

An HBGA utilizing MANET that is considered in this paper is one type of parallel GAs including migration of candidate solutions between different sub-populations. Especially, a concrete HBGA used in simulations in Section IV for basic evaluation of the concept of HBGA utilizing MANET is this type of parallel GA whose sub-populations consist of just one candidate solution. However, in an actual HBGA, H-agents determine their behaviors such as how they move, when they create new candidate solutions, and which candidate solutions they make migrate. This freedom does not exist in parallel GAs with C-agents. Furthermore, HBGA used in simulations presented in Section IV also differ from other parallel GAs in a point that in the HBGA, receivers of migrated candidate solutions judge whether to accept migrated candidate solutions based on the receivers' interest.

MANET assumed in this paper is a network consisting of H-agents that happen to meet and connect to each other by wireless connections. This kind of network is called an "opportunistic network". Then, we use "interest of people" for information (candidate solutions) sharing in an opportunistic network. Some existing methods for propagating information in an opportunistic network use "interest of people" as identifiers. That is to say, they utilize interest of people for judging whether to propagate information from a node to a node.

Haggle[12] is a software for information sharing that includes information propagation with human interests. Here, information sending and receiving is conducted based on human interests represented by texts. In [13], an information propagating and sharing method with human interests has also been proposed. The method represents human interests by real-valued vectors whose elements take a real number between 0 and 1. In this method, nodes exchange their interests with each other locally, and then give information matching their interests to each other during their communication time. The number of neighboring nodes with which a node exchanges information is a parameter of the method. The information propagating and sharing method presented in this paper also use real-valued vectors as representations of human interests.

In addition, there are several studies on information sharing in an opportunistic network that considers human interests [14][15]. In [14], assuming that two moving nodes meet and share information matching their interests if they have such information, the impact of density of nodes and diversity of human interests on the results of the information sharing is investigated. The focus in [15] is a relationship between presence or absence of a spirit of cooperation in nodes in terms of information propagation among the nodes and the number of times of obtaining information. In [15], an information sharing method with human interests is also considered, and the method adaptively decides whether a node propagates information for others, that is, whether a node cooperates with others, using interests of its surrounding nodes (humans).

III. Organizing Human Agents in MANET

First of all, here we assume a MANET that is formed by moving H-agents with wireless communication devices. In fact, there are protocols to form MANET and enable communication between particular wireless communication devices, but we do not consider the details of the protocols. We assume that closeness of devices is sufficient for ensuring communication between devices, independent of the particular communication protocol that is used. Especially, to enable communication between particular devices via several other devices, a routing protocol is needed. In this paper, we assume that communication devices communicate via broadcasting, which means that a focus device sends out information to all devices in its communicable area. In addition, we assume that a problem to be solved by HBGA is shared among H-agents by means of broadcasting.

A. Features of HBGA in MANET

To clarify features of distributed HBGA conducted in MANET, we compare the distributed HBGA with centralized HBGA that manages candidate solutions in a centralized manner such that candidate solutions are shown on a Web message board.

1) *Candidate solutions are propagated to others "indirectly" or "directly".*
   In the centralized HBGA utilizing Web etc., propagation of candidate solutions to others is always conducted via a mechanism for managing all candidate solutions. The candidate solutions managed there become individuals of the centralized HBGA. Meanwhile, the distributed HBGA utilizing MANET does not have a mechanism for managing all candidate solutions. Candidate solutions are shared only among H-agents who can communicate with each other. Therefore, the final candidate solution is surely shared among all H-agents in the centralized HBGA, but might be shared only among some H-agents in the distributed HBGA.

2) *Genetic operators are executed "at one place" or "at many places".*
The centralized HBGA has a place to manage all candidate solutions, for example, a Web message board for solving a problem, and executes genetic operators such as selection, crossover, and mutation at the place by H-agents. Meanwhile, in the distributed HBGA utilizing MANET, for all H-agent, their own wireless communication device used for receiving others' candidate solutions become locations to execute genetic operators by themselves.
3) *H-agents to whom candidate solutions can be notified are "physically changed" or "not physically changed".*
The centralized HBGA can notify any candidate solutions to all H-agents using the mechanism for managing them. Therefore, we can say that H-agents to whom candidate solutions are notified are theoretically not changed over time. Meanwhile, in the distributed HBGA utilizing MANET, since each H-agent has a limited communication range and moves, H-agents to whom candidate solutions are notified are changed over time.

From the features of HBGA utilizing MANET described above, we can see that considering how to organize H-agents is equivalent to considering how to share candidate solutions among moving H-agents with limited communication ranges.

B. Proposed Method for Organizing Human Agents

The method for organizing H-agents proposed here determines who propagates his/her candidate solution to whom. Since the fundamental way for the propagation in the proposed method is broadcasting, receivers of broadcasted candidate solutions judge if they actually accept them. As mentioned below, the judgment is done probabilistically.

A H-agent refers to candidate solutions received from other H-agents when the H-agent is going to create a new candidate solution. A procedure in which a H-agent creates a new candidate solution from her/his own candidate solution and the referred candidate solutions of others is correspondent to selection, crossover, and mutation operators in GA. This procedure is done by H-agents, and is not included in the proposed method for organizing H-agents.

The proposed method uses identifiers of H-agents that represent interests or preferences of the H-agents on realized solutions, such that a H-agent is interested in "cost" of realized solutions or interested in "safety" of realized solutions. We will explain this point below.

1) Identifiers Representing Human Interests: An identifier representing human interests on realized solutions of a given problem is an L-dimensional real-valued vector. L is a parameter of the method presented in this paper and each element of the vector takes a real value within $[0, 1]$. Each element corresponds to a specific object of interest to humans on realized solutions. For example, an interest in "cost" of realized solutions corresponds to a certain element of the L elements. An element value of 0 in the vector indicates that the H-agent with the identifier is not interested in the corresponding object of interest. Conversely, an element value of 1 indicates a maximum interest in the corresponding aspect. Real values between 0 and 1 represent the degree of interest in a specific object. H-agents involved in HBGA can create and modify the identifiers representing their interests at any time.

C. Forwarding Solution Candidates Using Distances between Identifiers

H-agents who form MANET publish and forward their candidate solutions to their surroundings. In this candidate solutions publishing and forwarding, the identifier representing the interests of a H-agent who wants to publish her/his candidate solution is assigned to the candidate solution that the H-agent actually publishes. Let $\boldsymbol{s}= (s_1, s_2, \ldots, s_L)$ be this identifier of the published candidate solution, which is the same as the identifier of the H-agent. Forwarding the published candidate solution is conducted by broadcasting.

H-agents who have received the published candidate solution from their neighbors determine whether they should further forward the received candidate solution to their surroundings by broadcasting that uses the identifier assigned to the published candidate solution, $\boldsymbol{s}$ and the identifier of the H-agent who has received the candidate solution, $\boldsymbol{d}= (d_1, d_2, \ldots, d_L)$ (see Figure 1). Specifically, the H-agent who has received the candidate solution forwards the received candidate solution to neighbors with probability P_b, expressed by Equation (1), where the candidate solution is eliminated by the H-agent with probability $1 - P_b$.

$$P_b = \exp\left(-d \times \frac{\sum_{i=1}^{L}(s_i - c_i)^2}{L}\right). \tag{1}$$

where $s_i, c_i \in [0, 1] \subset \boldsymbol{R}$, $\frac{\sum_{i=1}^{L}(s_i-c_i)^2}{L} \in [0, 1]$, and d is a parameter of the proposed method. Figure 2 shows plots of P_b for different values of d. Equation (1) indicates that the larger a distance between identifiers, the smaller the propagation probability P_b. When H-agents receive the same candidate solution that they have already received once in the present series of candidate solution forwarding by broadcast, they immediately eliminate the received candidate solution. Therefore, they do not judge whether to forward the candidate solution according to the probability expressed by Equation (1) more than once in a series of candidate solution forwardings.

IV. Simulations

A. Simulation Model and Scenario

The simulation model used assumes a 10×10 square area without obstacles, in which three hundred H-agents move around. The position of a H-agent takes real values like $(2.2, 7.5)$. If a H-agent is about to cross the border of the square area, the border reflects the H-agent regularly. Each H-agent randomly moves to a position within a 1×1 square area centered on his/her current position, once per unit of time. Each H-agent is assumed to be able to communicate with other H-agents within a circle of radius 1 from the H-agent.

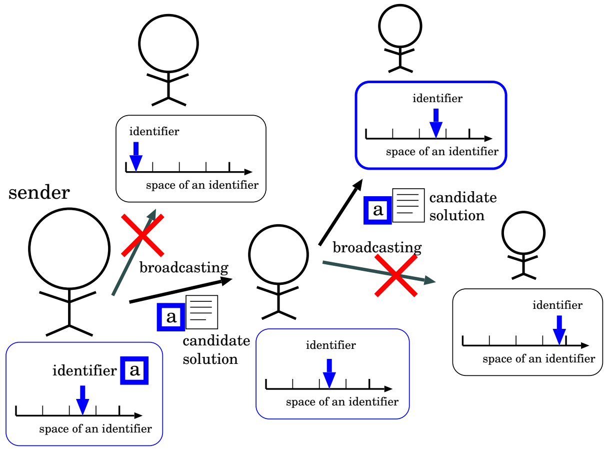


Fig. 1. Forwarding a candidate solution.

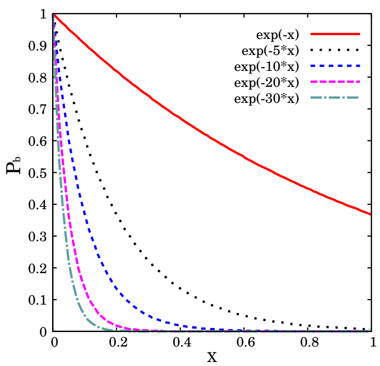


Fig. 2. Plots of P_b for different values of d, which is the probability with which a H-agent who received candidate solution forwards that received candidate solution to neighbors by broadcast. The probability with which the candidate solution is eliminated by the H-agent is $1 - P_b$. In this figure, $X = \frac{\sum_{i=1}^{L}(s_i - c_i)^2}{L}$.

The simulation scenario considered here is prepared for evaluations of the proposed method used in the HBGA utilizing MANET, and is to execute a function minimization by the HBGA utilizing MANET. There are three types of functions for minimization, which are shown in Table I. The number of decision variables, n, is 20 for all the functions.

B. Configurations of HBGA

It is assumed that all 300 H-agents execute candidate solution forwarding among the H-agents once per unit of time. Then, every H-agent creates new candidate solutions based on her/his present candidate solution and others' candidate solutions accepted per unit of time. Thus, in the distributed HBGA considered here, all H-agents become creators of new candidate solutions as well as evaluators of created candidate solutions.

A new candidate solution is created by applying BLX-α ($\alpha = 0.36$) crossover operator to a pair of the present candidate solution of a H-agent and each of received candidate solutions of others, and then by applying a mutation operator that randomly changes a value of each decision variable of the created candidate. The crossover and mutation rates are 1.0 and 0.005, respectively. In such a way to create a new candidate solution, at each H-agent, new candidate solutions are created. The number of created candidate solutions is equal to the number of others' candidate solutions that the H-agent accepted. Then, those new candidate solutions created at each H-agent are evaluated by the H-agent and only the new candidate solution with the best fitness value among all the created candidate solutions is held by the H-agent as her/his candidate solution of next time unit. Here, the maximum number of others' candidate solutions that each H-agent can receive is set to be 60. Each H-agent stops receiving candidate solutions from others when the number of received candidate solutions reaches 60.

As mentioned above, at every time unit, each H-agent holds the candidate solution with the best fitness value among those that the H-agent created so far. Here, we use this candidate solution of each H-agent as an identifier of H-agent representing interests of the H-agent used in the method for organizing H-agents described in Section III-B. Using the present candidate solutions as the identifiers is not original intention, but a position in a search space is set to be the identifier in this paper. In addition, we use $10, 20, 30$ for the parameter d in Equation (1), which is used for judging whether to forward a candidate solution in the method for organizing H-agents.

C. Simulation Results

Simulation results are shown in Table II. In Table II, results of real-coded GA for comparison are also shown, which uses Minimal Generation Gap as a generation gap model, BLX-α ($\alpha = 0.36$) as a crossover operator, and a uniform-random-number-base mutation operator. The crossover and mutation rates for this real-coded GA are 1.0 and 0.005, respectively, which are the same for the HBGA used here.

We can observe from Table II that the HBGA using the proposed method for organizing H-agents yields a little higher search performance as the value of d in Equation (1) for forwarding candidate solutions increases, for all the three types of fitness functions used. Especially, for the Schwefel function, when using $d = 30$, the search performance is much improved compared to the cases of using the other values of d. In the simulations here, we set the identifier of each H-agent used for forwarding a candidate solution to be the same as the best candidate solution created so far. So, larger values of d cause a situation where sharing candidate solutions is facilitated only among H-agents whose identifiers are close to each other, and in other words, application of the crossover operator is very likely to occur only among candidate solutions located in the same local area in the search space. It would facilitate to improve quality of candidate solutions.

The Rosenbrock function is a unimodal function in all the fitness function used. Since it is unimodal, the HBGA is easy to gather candidate solutions around the global optimum, and then this contributes to acceleration of sharing

TABLE I

FITNESS FUNCTIONS USED FOR EVALUATIONS OF THE PROPOSED METHOD. THE VALUES OF n IS 20 IN THE SIMULATIONS.

Label	Function	Domain	Optimum solution	Optimum value
Rosenbrock	$\sum_{i=2}^{n}\{100(x_1 - x_i^2)^2 + (x_i - 1)^2\}$	$x_i \in [-2.048, 2.048]$	$\forall i\ x_i = 1\ \ (i = 1, 2 \ldots, n)$	0
Rastrigin	$10n + \sum_{i=1}^{n}\{x_i^2 - 10\cos(2\pi x_i)\}$	$x_i \in [-5.12, 5.12]$	$\forall i\ x_i = 0\ \ (i = 1, 2 \ldots, n)$	0
Schwefel	$418.9829n + \sum_{i=1}^{n}\{-x_i \sin\sqrt{\lvert x_i\rvert}\}$	$x_i \in [-500, 500]$	$\forall i\ x_i = 420.9687\ \ (i = 1, 2 \ldots, n)$	0

TABLE II

SIMULATIONS RESULTS. THE RELATIONSHIP BETWEEN THE NUMBER OF FITNESS EVALUATIONS AND FITNESS VALUES IS SHOWN. THE FITNESS VALUE IS CALCULATED OVER 50 INDEPENDENT SIMULATIONS RUNS.

the number of fitness evaluations	1×10^3	1×10^4	5×10^4	1×10^5	2×10^5	3×10^5	4×10^5
Real-coded GA							
Rosenbrock	7.257×10^2	2.953×10^1	9.457×10^0	8.538×10^0	7.329×10^0	6.500×10^0	5.866×10^0
Rastrigin	2.112×10^2	1.026×102	1.179×10^1	3.274×10^0	1.666×10^{-7}	0.000×10^{-10}	0.000×10^{-10}
Schwefel	5.766×10^3	4.215×10^3	7.617×10^2	5.196×10^1	5.546×10^{-8}	5.513×10^{-8}	5.513×10^{-8}
HBGA ($d = 10$)							
Rosenbrock	8.523×10^2	1.043×10^2	2.164×10^1	1.280×10^1	7.771×10^0	6.930×10^0	6.750×10^0
Rastrigin	2.129×10^2	1.292×10^2	6.203×10^1	3.437×10^1	1.520×10^1	9.247×10^0	6.723×10^0
Schwefel	5.812×10^3	5.018×10^3	3.351×10^3	2.227×10^3	1.056×10^3	5.965×10^2	3.480×10^2
HBGA ($d = 20$)							
Rosenbrock	6.368×10^2	5.483×10^1	1.841×10^1	1.121×10^1	7.385×10^0	6.825×10^0	6.554×10^0
Rastrigin	2.041×10^2	1.081×10^2	4.307×10^1	2.266×10^1	1.214×10^1	8.842×10^0	6.972×10^0
Schwefel	5.732×10^3	4.191×10^3	1.601×10^3	7.172×10^2	2.134×10^2	5.863×10^1	1.165×10^1
HBGA ($d = 30$)							
Rosenbrock	6.125×10^2	4.697×10^1	1.770×10^1	1.083×10^1	7.589×10^0	6.637×10^0	6.449×10^0
Rastrigin	1.931×10^2	8.162×10^1	3.091×10^1	1.844×10^1	1.059×10^1	7.800×10^0	6.084×10^0
Schwefel	5.649×10^3	3.378×10^3	1.099×10^3	6.077×10^2	1.617×10^2	7.760×10^0	1.556×10^{-3}

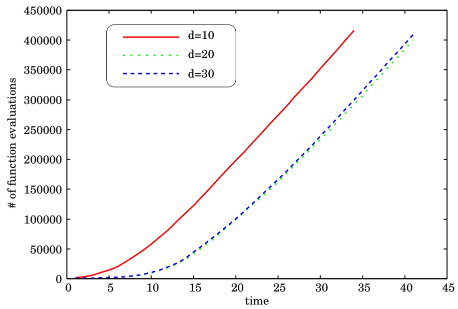
and creating candidate solutions among the H-agents. Table II shows that there is no significant difference in the search performance between the HBGA and the real-coded GA. Both of the methods cannot make candidate solutions approach to the global optimum whose fitness value is 0, because the used crossover operator does not handle correlations between decision variables included in the fitness function well.

Next, the Rastrigin function is a multimodal function that has a big valley structure whose ridgeline is unimodal-shaped and does not include correlations between decision variables. Since this function has multi-modality as well as a big valley structure, it is easy for the HBGA to move candidate solutions in the direction of the global optimum, and then it is expected that this contributes to acceleration of sharing and creating candidate solutions among the H-agents. However, Table II shows that the HBGA is much inferior to the real-coded GA for comparison. That would be because most candidate solutions separately held by the H-agents are trapped in local optima around the global optimum. If all candidate solutions can be used for exploiting the global optimum in a centralized manner as in the real-coded GA for comparison, the candidate solutions would be improved.

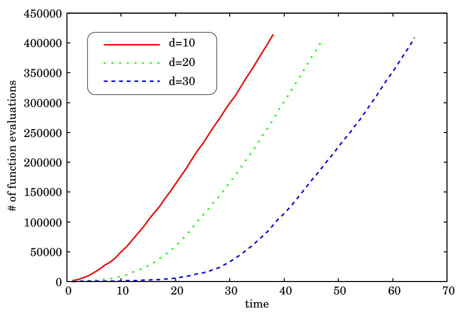
Finally, Schwefel function is a multimodal function without correlation between decision variables. The best and the second best solutions in this function are distant from each other in the search space. Therefore, it should be hard for the HBGA to gather solution candidates around the global optimum. Table II shows that the HBGA is much inferior to the real-coded GA for comparison except when $d = 30$.

As mentioned above, we saw that the difference in the search performance between the HBGA and the real-coded GA for comparison depends on the used fitness function. To discuss the reason for this fact, we show the relationship between time and the number of function evaluations for each fitness function in one simulation run in Figure 3. Every H-agent publishes her/his created candidate solution to the surroundings, but the number of others' candidate solutions that each H-agent receives depends on its identifier. Therefore, the number of fitness evaluations for candidate solutions created between the present candidate solution of the H-agent and each of the received candidate solutions of others varies with time. Figure 3 shows that for any fitness function, the number of function evaluations increases with the time. The increase rate of the number of function evaluations is higher in the order of Rosenbrock, Rastrigin, and Schwefel. This order seems related to the uni-modality of the fitness function used. In addition, we can observe from Figure 3 that the time for the number of function evaluations to notably increase becomes later as the value of d becomes larger. The HBGA with smaller value of d would lose diversity of the candidate solutions separately held by the H-agents more quickly than that with large value of d, and therefore, the HBGA yields a little higher search performance as the value of d increases, as mentioned above.

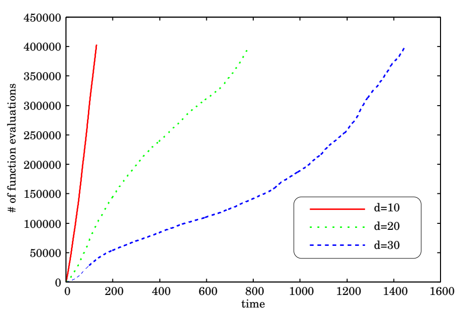
As for Schwefel function, in the case of $d = 30$, the tendency of increase of the number of function evaluations is much different from the other cases. The increase rate of the number of function evaluations in the early stage of the search is quite low, but in the later stage, the increase rate gets



(a) Rosenbrok.



(b) Rastrigin.



(c) Schwefel 関数.

Fig. 3. The relationship between time and the number of function evaluations for each fitness function in one simulation run of the HBGA.

remarkably higher. Looking at the convergence characteristics shown in Table II, we can see that the candidate solutions approach to the global optimum in this case. This would be because in the case of $d = 30$, the diversity of candidate solutions is well maintained in the early stage of the search, that is, not all candidate solutions gather around the second best solution. Then, the candidate solutions would successfully converge to the global optimum.

In summary, the HBGA including the proposed method for organizing H-agents is inferior to the real-coded GA for comparison in terms of the search performance, but we can confirm that the HBGA has ability in converging the candidate solutions. Therefore, we conclude that the basic usefulness of the proposed method for organizing H-agents is shown.

V. Concluding Remarks

In the present paper we proposed the method for organizing human agents in HBGA utilizing MANET, and demonstrated the basic usefulness of the proposed method through simulations. We introduced simplified models into the simulations to examine the proposed method, which were related to behaviors of human agents such as moving, creating candidate solutions, and forwarding their candidate solutions, as well as identifiers of human agents that are used for sharing candidate solutions. We will consider more realistic models and examine the proposed method further.

References

[1] A. Kosorukoff,"Human based genetic algorithm", 2001 IEEE International Conference on Systems, Man, and Cybernetics, pp.3464-3469, 2001.
[2] D. E. Goldberg, "The design of innovation: Lessons from and for competent genetic algorithms", Kluwer Academic Publishers, USA, 2002.
[3] H. Takagi, "Interactive evolutionary computation: fusion of the capacities of EC optimization and human evaluation", Proc. of the IEEE, 2001.
[4] Free Knowledge Exchange (FKE) project, "http://3form.org".
[5] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, "Mobile Ad Hoc Networking", Wiley-IEEE Press, 2004.
[6] S. Kumar, R. K. Rathy, and D. Pandey, "Design of an ad-hoc network model for disaster recovery scenario using various routing protocols", Proceedings of the International Conference on Advances in Computing, Communication and Control (ICAC3 '09), pp.100-105, 2009.
[7] K. Shimoda and K. Gyoda, "Analysis of Ad Hoc Network Performance for Disaster Communication Models", Proceedings of 10th International Symposium on Autonomous Decentralized Systems (ISADS 2011), pp.483–488, 2011.
[8] A. Kosoruko and D. E. Goldberg, "Evolutionary computation as a form of organization", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002) pp.965–972, 2002.
[9] C. D. Cheng and A. Kosorukoff, "A. Interactive one-max problem allows to compare the performance of interactive and human-based genetic algorithms", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004), pp. 983-993, 2004.
[10] X. Llora, K. Ohnishi, Y.-p. Chen, D. E. Goldberg, and Michel E. Welge, "Enhanced innovation: A fusion of chance discovery and evolutionary computation to foster creative processes and decision making", Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004), pp.1314–1315, 2004.
[11] G. Luque and E. Alba, "Parallel Genetic Algorithms: Theory and Real World Applications", Springer-Verlag, 2011.
[12] Haggle, "http://code.google.com/p/haggle/".
[13] I. Carreras, F. D. Pellegrini, D. Miorandi, D. Tacconi, and I. Chlamtac, "Why neighbourhood matters: interests-driven opportunistic data diffusion schemes", Proceedings of the third ACM workshop on Challenged networks, pp.81–88, 2008.
[14] W. H. Yuen, S.-C. Mau, and R.D. Yates, "Existence of data and multiuser diversities in noncooperative mobile infostation networks", IEEE Transactions on Mobile Computing, vol.8, no.8, pp.1117–1131, 2009.
[15] S. M. Allen, M. J. Chorley, G. B. Colombo, and R.M. Whitaker. Self adaptation of cooperation in multi- agent content sharing systems. Proceedings of Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2010), pages 104–113, 2010.
[16] L. J. Eshelman and D. J. Shaffer, "Real-coded genetic algorithms and interval-schemata", Foundations of Genetic Algorithms 2, pp.187–202, 1993.
[17] H. Satoh, M. Yamamura, and S. Kobayashi, "Minimal generation gap model for GAs considering both exploration and exploitation", Proceedings of the International Conference on Fuzzy Systems, Neural Networks and Soft Computing (Iizuka'96), pp.494–497, 1996.