2014 IEEE International Conference on Systems, Man, and Cybernetics
October 5-8, 2014, San Diego, CA, USA

Team of Bayesian Optimization Algorithms to Solve Task Assignment Problems in Heterogeneous Computing Systems

Jie Li[1], JunQi Zhang[1,†], Qi Kang[2], *Member, IEEE* and ChangJun Jiang[1], *Member, IEEE*
[1]Department of Computer Science and Technology, Tongji University, Shanghai, China
Key Laboratory of Embedded System and Service Computing, Ministry of Education
[2]Department of Control Science and Engineering, Tongji University, Shanghai, China
† Corresponding author. Email: lijietjsh@gmail.com, {zhangjunqi, qkang, cjjiang}@tongji.edu.cn

***Abstract*—A Bayesian optimization algorithm (BOA) belongs to estimation of distribution algorithms (EDAs). It is characterized by combining a Bayesian network and evolutionary algorithms to solve nearly decomposable optimization problems. BOA is less popularly applied to solve high dimensionality complex optimization problems. A key reason is that the cost of training all dimensions by BOA becomes expensive with the increase of problem dimensionality. Since data are relatively sparse in a high dimensional space, even though BOA can train all dimensions simultaneously, the interdependent relations between different dimensions are difficult to learn. Its search ability is thus significantly reduced. In this paper, we propose a team of Bayesian optimization algorithms (TBOA) to search and learn dimensionality. TBOA consists of multiple BOAs, in which each BOA corresponds to a dimension of the solution domain and it is responsible for the search of this dimension's value region. The proposed TBOA is used to solve the real problem of task assignment in heterogeneous computing systems. Extensive experiments demonstrate that the computational cost of the overall training in TBOA is decreased very significantly while keeping high solution accuracy.**

***Keywords*—Estimation of distribution algorithm (EDA), Bayesian optimization algorithm (BOA), dimensionality learning.**

I. Introduction

An estimation of distribution algorithm (EDA) is an advanced evolutionary algorithm first presented in [1]. Its most significant characteristic is that it can explicitly build the distribution of promising solutions and estimate their correlations. It has been successfully applied to solve various problems, e.g., multiobjective multiple traveling salesman [2], maximum diversity [3] and multiobjective resource-constrained project [4].

A Bayesian optimization algorithm (BOA) [5] belongs to EDAs. Its estimation and sampling are based on a constructed Bayesian network, which is designed to automatically estimate and construct the joint probability distribution of promising solutions in a search space. New candidate solutions are generated by sampling this Bayesian network and then they are incorporated into the current population to eliminate some relatively worse solutions. As an important evolutionary algorithm, BOA has been widely employed to solve all kinds of nearly decomposable optimization and scheduling problems, such as electric equipment configuration problem in a power plant [6], decomposition problems [7], fault identification on flight control system [8], nurse scheduling [9] and directed acyclic graph (DAG) scheduling [10]. The work [11] presents a hierarchical Bayesian optimization algorithm (hBOA) that can solve hierarchical problems accurately and reliably by exploiting hierarchical decomposition as opposed to decomposition in a single level. The work [12] presents a mixed Bayesian optimization algorithm (MBOA) that tries to learn a Bayesian network via a set of decision trees with univariate normal-kernel leaves. The work [13] presents a real-coded Bayesian optimization algorithm (rBOA) that uses a Bayesian network with a mixture of normal distributions sampling to solve continuous optimization problems.

Although the aforementioned works have obtained satisfactory results, there is still a drawback in BOA. It is only used to deal with optimization problems with low dimensionality, and is rarely applied for solving high dimensionality complex optimization problems. The most important reason is that as the problem dimensionality increases, a BOA with a single Bayesian network will spend enormous cost to train all dimensions. Moreover, in a high dimensional space, even though a BOA can train all dimensions at the same time, the interdependent relations between different dimensions cannot be effectively represented because of the relatively sparse data issue. The search ability of BOA can thus be decreased dramatically.

How to provide an effective algorithm that can adapt to multi-dimensional problems without reducing its search ability becomes an important and interesting problem. This work for the first time proposes a team of BOAs (TBOA) to search and learn dimensionality. Each BOA only corresponds to a dimension and it is responsible for the search of this dimension's value region. It is divided into different parts, each of which corresponds to a continuous interval if it is continuous or to an integer value if it is discrete. Each BOA undertakes dimensional search and can estimate the joint probability distribution of different values that belong to this dimension, thereby leading to its own Bayesian network.

Compared with the previous works, TBOA has two significant characteristics:

1) It can obtain better solutions than traditional BOA via dimensional search precisely; and

978-1-4799-3840-7/14/$31.00 ©2014 IEEE

2) It needs less computational cost of the overall training since the Bayesian network of each BOA in TBOA is only responsible for the training of a dimension, while in traditional BOA, a Bayesian network needs to train all dimensions simultaneously.

The proposed TBOA is applied to solve a task assignment problem in heterogeneous computing (HC) systems. The goal of this problem is to find an appropriate allocation that minimizes the total costs of execution and communication without violating any of the constraints. Extensive experiments are carried out on this problem to demonstrate the superiority of TBOA over BOA. The experimental results illustrate that the computational cost of the overall training in TBOA is dramatically decreased by more than 80% while keeping high solution accuracy.

Next, a task assignment problem is formulated in Section II. In Section III, BOA algorithm is introduced. The proposed TBOA is presented in Section IV. In Section V, it is applied to solve task assignment problems. Extensive simulation results comparing TBOA and BOA are presented in Section VI. Finally, concluding remarks are given in Section VII.

II. Task assignment problem formulation

A task assignment problem in HC systems has the following characteristics: a task interaction graph (TIG) $G(V, E)$ is used to represent a distributed application, where V is a set of N tasks and E is a set of edges that define the communication requirements among N tasks. Precedence relations among tasks are not considered. A weight w_{ij} represents the amount of data to be transferred between tasks i and j. The estimated execution cost matrix $H = [y_{ik}]_{N\times M}$ where y_{ik} is the cost to execute task i on processor k, N is the number of tasks, and M is the number of processors. Next, d_{hq} is defined as communication cost associated with one unit of data transferred from processors h to q. It is symmetric, i.e., $d_{hq} = d_{qh}$. Here, different communication cost is incurred if an identical amount of data is transmitted through different communication channels. Thus, $w_{ik}d_{hq}$ defines a communication cost if tasks i and k are executed on processors h and q, respectively.

The resource requirement by the i-th task, denoted as r_i, and the available resource capacity of the p-th processor, denoted as O_p constitute the constraints. The task assignment problem in HC systems can be formulated as:

$$min\ cost(\theta) = \sum_{i=1}^{N} y_{i\theta[i]} + \sum_{i=1}^{N-1}\sum_{k=i+1}^{N} w_{ik}d_{\theta[i]\theta[k]}, \forall\theta \in \vartheta \quad (1)$$

$$s.t. \sum_{i:\theta[i]=p} r_i \le O_p\ p \in \{1, 2, \ldots, M\} \quad (2)$$

where θ is an integer vector that represents the assignment of a particular task and ϑ is the set of all mappings. $\forall\theta \in \vartheta$, $\theta[i] = p$ denotes task i being allocated to processor p. Eq. (1) consists of two parts. The first part is the sum of the execution costs and the second part is the sum of the communication costs among interacting tasks located at different processors. The scheduling goal is to minimize the sum of the two parts' costs. Eq. (2) is a constraint that ensures the total resource requirements of those tasks assigned to each processor must not exceed its resource availability.

III. BOA

BOA [5][11] belongs to a class of EDAs. It is based on the estimation and sampling of a constructed Bayesian network, which is designed to automatically encode promising solutions' structure and estimate their joint probability distribution. New offspring are generated by sampling the constructed Bayesian network and are incorporated into the population such that some worse solutions can be eliminated. **Algorithm 1** is a description of BOA.

Algorithm 1 BOA

1: Generate the initial population randomly;
2: Select some promising solutions from the current population using a certain selection procedure, e.g., tournament selection or truncation selection;
3: Construct a Bayesian network by estimating these selected promising solutions;
4: Sample new offspring according to the joint probability distribution encoded by this Bayesian network;
5: Use new offspring to replace some worse solutions in the previous population;
6: If the end condition is not satisfied, go to Step 1.

Bayesian network [14] is a directed acyclic graph (DAG) that is composed of nodes, edges and the conditional probability table (CPT) that can identify the conditional dependencies. Bayesian network can be used to calculate the joint probability distribution among different nodes. It can be decomposed into the production of the marginal distribution randomly, that is:

$$P(x_1, x_2, \ldots|\mathbf{B}) = \prod_{i=1}^{L} P(x_i, \delta_i) = \frac{\sum_{A-\{x_i\}\bigcup\delta_i} P(x_1, x_2, \ldots)}{\sum_{A-\{x_i\}} P(x_1, x_2, \ldots)} \quad (3)$$

where $x_1, x_2, \ldots$ represent the set of nodes in Bayesian network, L is the number of nodes. $\mathbf{B}$ is denoted as a Bayesian network, $\delta_i : i = 1, 2, \ldots$ means the set of the i-th node's parent nodes.

IV. Proposed TBOA

The significant characteristic of BOA is that it owns an excellent estimation characteristic through a Bayesian network. However, it is difficult to estimate the interdependent relations between different dimensions as problem dimensionality increases. Inspired by swarm intelligence approaches [15] and clustering [16], this work proposes an effective algorithm to search and learn dimensionality, called a team of BOAs (TBOA). It consists of multiple BOAs, in which the number of BOAs used is equal to problem dimension count. Each BOA is responsible for searching in a dimension. It is used to search this dimension's value region, then a Bayesian network is constructed by estimating the interdependent relationships between different values and building the joint probability distribution of different values. Next, each BOA uses this Bayesian network to sample some new values. In the end, new offspring of problem are composed by all BOAs' sampling values.

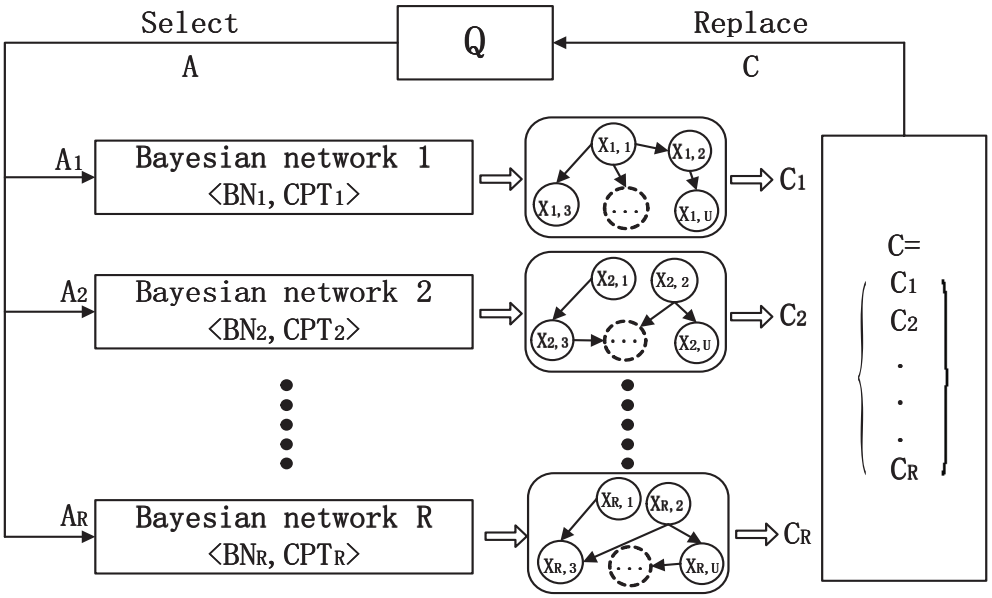


Fig. 1. The structure of TBOA where Q is current population.

A. Structure of TBOA

TBOA consists of multiple BOAs. Each BOA is responsible for searching in a dimension, and dimensional values in this dimension are sampled by its Bayesian network. Fig. 1 illustrates its structure. It consists of R BOAs. R is an optimization problem dimension count. Each BOA is considered as an independent entity and responsible for searching in a dimension. The Bayesian network of the i-th BOA can be defined as $< BN_i, CPT_i >, 1 \leq i \leq R$, in which BN_i is the i-th Bayesian network structure, and CPT_i is as conditional probability table. $x_{i,1}, x_{i,2}, \ldots, x_{i,U}$ denote the values in the i-th Bayesian network, U is the number of values in the i-th dimension's value region. Each value corresponds to a continuous interval if the value region is continuous or corresponds to an integer value if it is discrete. A_i denotes the selection states of all values according to those selected promising solutions that correspond to the i-th dimension, C_i is the new values sampled from the i-th Bayesian network. $C = \{C_1, C_2, \ldots, C_R\}$ is the new offspring set that sampled from all Bayesian networks. "Select" denotes some promising solutions are selected from the current population according to their fitness values via a certain selection procedure. "Replace" denotes some worse solutions in the population are replaced by new offspring.

Next, we describe the process of TBOA. First, a promising set A of solutions is selected from the current population Q. Then A is mapped into $A_i, 1 \leq i \leq R$ that denotes selection states of all values in the i-th dimension according to those selected promising solutions. Next, BOA i uses its A_i to estimate the joint probability distribution of different values and construct the i-th Bayesian network, and then obtains the interdependent relationships between different values. The interdependent relationships indicate that whether to select a value is influenced by the selection states of this value's parents. Third, each Bayesian network samples the joint probability distribution to obtain a certain number of new candidate values C_i, which are integrated into the new offspring set $C = \{C_1, C_2, \ldots, C_R\}$. At last, C is used to replace some worse solutions in population Q, as shown in Fig. 1.

B. Generation of Multiple Bayesian Networks

Now, we describe how to generate multiple Bayesian networks in TBOA. Each Bayesian network corresponds to a dimension that contains U values. First, some promising solutions are selected from the current population according to their fitness values via a selection procedure. Second, construct multiple Bayesian networks according to the selected promising solutions. The structure of each Bayesian network in TBOA is encoded by a directed acyclic graph with the nodes correspond to the values in this dimension's value region and the edges correspond to their conditional dependencies. In order to map the selected promising solutions to each dimension, we define a new data set as follows. Let an $S \times R$ matrix Z denote the selected promising solutions, in which S is the number of promising solutions, and R is the number of dimensions.

$$Z = \begin{bmatrix} z_{1,1} & \cdots & z_{1,R} \\ \cdots & z_{s,r} & \cdots \\ z_{S,1} & \cdots & z_{S,R} \end{bmatrix} \tag{4}$$

where $z_{s,r}$ indicates the promising value of the r-th dimension in the s-th promising solution, $1 \leq s \leq S, 1 \leq r \leq R$. Next, Z is decomposed and converted to a new matrix set A^r, in which each matrix corresponds to a dimension of the solution domain.

$$A^r = \{ \begin{bmatrix} a^1_{1,1} & \cdots & a^1_{1,S} \\ \cdots & a^1_{u,s} & \cdots \\ a^1_{U,1} & \cdots & a^1_{U,S} \end{bmatrix} \cdots \begin{bmatrix} a^R_{1,1} & \cdots & a^R_{1,S} \\ \cdots & a^R_{u,s} & \cdots \\ a^R_{U,1} & \cdots & a^R_{U,S} \end{bmatrix} \} \tag{5}$$

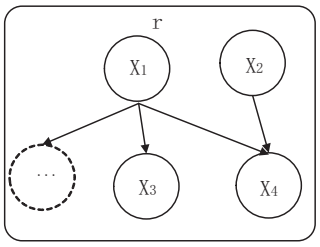
where $a^r_{u,s}$ indicates the u-th value's selection state of the r-th dimension in the s-th promising solution. $1 \leq u \leq U$. Matrix

set A^r is used to estimate and construct Bayesian networks in TBOA. Each element $a^r_{u,s}$ has only two values, i.e., $a^r_{u,s} \in \{0, 1\}$. "1" means the r-th dimension selects the u-th value in the s-th promising solution, while "0" means "not".

C. Sampling and Analysis of Each Bayesian Network

The sampling process in TBOA is different from traditional BOA. In traditional BOA, sampling is carried out according to the interdependent relationships between different dimensions, then a new offspring is generated by the dimensions order. In TBOA, sampling is carried out according to the interdependent relationships between different values in each dimension, so each dimension can sample its values independently, then a new offspring is generated by the combination of the values of different dimensions. The process of sampling in each BOA consists of two steps. The first step computes an ancestral ordering of the values, where each value is preceded by its parents. The basic idea is that the selection states of the parents of each value are generated prior to the generation of the selection state of the value itself. In the second step, the selection states of all values are generated according to the ancestral ordering. Assuming that we have a vector $\mathbf{b}_r$ for the r-th dimension at each sample, which is described as $\mathbf{b}_r = (b^1_r, \ldots, b^U_r)$. Every element in $\mathbf{b}_r$ indicates whether this dimension selects the corresponding value. "1" means yes, while "0" no. Note that all elements in $\mathbf{b}_r$ may be "1", which means all values are selected by this BOA sampling. A preservation mechanism can be used to preserve those desired values, e.g., preserving the first sampled "1" value or random preservation.

Next, we will briefly analyze the characteristics of each Bayesian network in TBOA. We assume that the structure of the r-th Bayesian network is shown in Fig. 2(a). It owns four values. At the initial iteration, the conditional probabilities between these values are similar, as described in the conditional probability table of Fig. 2(b). Sampling it may yield a vector $\mathbf{b}_r = (1, 1, 1, 1)$, which means the diversity of values is very high. It has a broad selection region and owns a high global search ability. With the increased number of iterations, TBOA may converge. The conditional probabilities among the four values also converge, as described in the conditional probability table of Fig. 2(c). Bayesian network r may be sampled to generate a new vector $\mathbf{b}_r = (0, 0, 0, 1)$, which means the diversity is very low, implying its high local search ability.



(a)

Bayesian network structure

X_1	X_2	$P(X_3=1\|X_1)$	$P(X_4=1\|X_1,X_2)$
0	0	0.55	0.65
0	1	0.55	0.45
1	0	0.60	0.55
1	1	0.60	0.70

(b)

X_1	X_2	$P(X_3=1\|X_1)$	$P(X_4=1\|X_1,X_2)$
0	0	0.20	0.65
0	1	0.20	0.25
1	0	0.30	0.20
1	1	0.30	0.15

(c)

Conditional probability table

Fig. 2. An example of sampling Bayesian network r. (a) Bayesian network structure and (b) (c) Conditional probability table.

V. Task Assignment in HC Systems

We now apply the proposed TBOA to solve a task assignment problem in HC systems. First, solution representation and how to generate initial population are given. Next, the structure of a Bayesian network is described. At last, a complete TBOA algorithm is represented.

A. Solution Representation and Initial Population

Solution representation is important for TBOA. A well-designed representation can make problem-solving easy. A potential solution for solving task assignment in HC systems is an integer vector $T[n], n \in \{1, 2, \ldots, N\}$ with N elements, $T[n] = m, m \in \{1, 2, \ldots, M\}$ denotes that processor m is assigned to the n-th task. N is the number of tasks and M is the number of processors. Initially, a population Q is randomly generated by a uniform distribution, the i-th dimension of the g-th solution is described as $x_{g,i} = randint(1, M)$, $i \in \{1, 2, \ldots, N\}$ and $g \in \{1, 2, \ldots, Q\}$. $randint$ is a function that generates an integer uniformly distributed in the integer range $[1, M]$. Then, the cost of a solution described in Eq. (1) is used as its fitness value.

B. Structures of Multiple Bayesian Networks

We use Eqs. (4) and (5) to construct the structures of multiple Bayesian networks for solving this task assignment problem. First, some promising solutions are selected from the current population according to their fitness values via a selection procedure. Next, the structures of multiple Bayesian networks in this task assignment problem are constructed according to Eqs. (4) and (5), where S is the number of promising solutions, N=R and M=U. Each matrix in Eq. (5) is used to estimate and construct a Bayesian network that searches a task assignment domain. Thus, each element in matrix $A^n_{m,s}$ can be "1" meaning that the n-th task is assigned on the m-th processor in the s-th promising solution, and"0" means "not".

C. Complete TBOA Algorithm

At each iteration, some new values (processors) are sampled in each dimension (task) according to the joint probability distribution encoded by the constructed Bayesian network. Note that there may be many 1's are sampled by a Bayesian network in a dimension. Implying that a task is assigned to different processors. This is unlikely to happen because a task can only be assigned to one processor in every assignment. This work adopts the first "1" sampled by a Bayesian network. After all tasks are assigned to their processors, a new offspring is generated. Thus, some new offspring are generated by all tasks sample their Bayesian networks repeatedly, then they replace some worse solutions in the previous population. The process of TBOA for solving a task assignment problem in HC systems is presented in **Algorithm 2**.

VI. Experimental Evaluation

Extensive simulations are carried out in order to compare TBOA with BOA [5]. They are coded in MATLAB-R2010a and dedicated simulations are executed on a 3.20GHz Core i3 processor with 4GB main memory running under Windows 7 environment.

TABLE I. EXPERIMENTAL RESULTS OVER 20 RUNS ON EACH OF 9 INSTANCES WITH 200 ITERATIONS.

Instances	BOA				TBOA					
	M_{avg}	V_{avg}	T_{avg}	B_{best}	M_{avg}	V_{avg}	T_{avg}	B_{best}	RPD_M	RPD_T
1	666.85	15.52	4.84+04	642.56	**653.67**	12.46	**1.29+03**	637.39	2.02	3.66+03
2	832.51	17.06	4.87+04	807.13	**797.81**	8.29	**1.26+03**	780.89	4.35	3.77+03
3	892.75	35.71	4.88+04	857.45	**855.11**	13.47	**1.29+03**	837.42	4.41	3.69+03
4	605.64	0	4.82+04	605.64	**604.90**	0.54	**1.31+03**	604.00	0.13	3.58+03
5	756.94	0	4.86+04	756.94	**729.08**	4.09	**1.28+03**	722.54	3.83	3.70+03
6	1270.55	15.58	3.54+04	1244.53	**1259.65**	17.67	**1.20+03**	1235.68	0.87	2.86+03
7	592.81	1.08	3.89+04	589.75	**585.61**	3.06	**1.22+03**	580.97	1.23	3.09+03
8	789.68	1.20-13	3.77+04	789.67	**770.40**	5.77	**1.20+03**	761.49	2.51	3.04+03
9	1434.40	5.03	3.94+04	1420.08	**1422.88**	18.06	**1.28+03**	1387.29	0.81	3.01+03

Algorithm 2 TBOA

1: At iteration $t \geq 1$, calculate the fitness value for each potential solution in the population;
2: Select two solutions from the current population randomly;
3: Compare the fitness values of the selected two solutions through a tournament selection procedure and the better one is retained. Then repeat Steps 2 and 3 until S promising solutions are selected;
4: Then Eqs. (4) and (5) utilize S promising solutions to construct multiple Bayesian networks;
5: Some new offspring are generated by all tasks sample their constructed Bayesian networks repeatedly;
6: S worse solutions in the current population are replaced by the top S new offspring regardless of their merits. Because these Bayesian networks are constructed by using promising solutions, most of new offspring are excellent;
7: If the end condition is not satisfied, go to Step 1.

The main parameters are N, M and D as the task interaction density that is a probability between two tasks. CCR is the communication to computation time ratio, which is the ratio of the average communication cost to the average computation cost. Three levels of task interaction density D=(0.3, 0.5 and 0.8) and three CCR=(0.5, 1.0 and 2.0) are tested, resulting a total of 9 benchmark instances. In our experiments, the values of (N, M) are set to (50, 5). The number of initial population for two algorithms is set to 100, S=50. In order to compare both algorithms fairly, a solution obtained by greedy constructive heuristic (GCH) algorithm [17] is integrated into the initial population. Their termination condition is that a maximum number of iterations, i.e., 200, is reached. Due to their stochastic nature, each of their independent runs may yield a different result. Thus, we run each algorithm 20 times for every benchmark instance and report the statistical results.

The experimental results are obtained by the corresponding algorithms to solve the problem instances, as shown in Table I. M_{avg} denotes the average total costs of execution and communication in Eq. (1), V_{avg} denotes the average standard deviation, T_{avg} denotes the average computational cost and B_{best} denotes the best total costs. The average relative percentage deviations of total costs RPD_M and computational cost RPD_T are calculated as follows:

$$RPD_M = (M_{BOA} - M_{TBOA})/M_{TBOA} \times 100 \quad (6)$$

$$RPD_T = (T_{BOA} - T_{TBOA})/T_{TBOA} \times 100 \quad (7)$$

where M_{TBOA} and T_{TBOA} are the average total costs and the average computational cost across 20 independent runs obtained by TBOA. M_{BOA} and T_{BOA} are the ones provided by BOA for each instance.

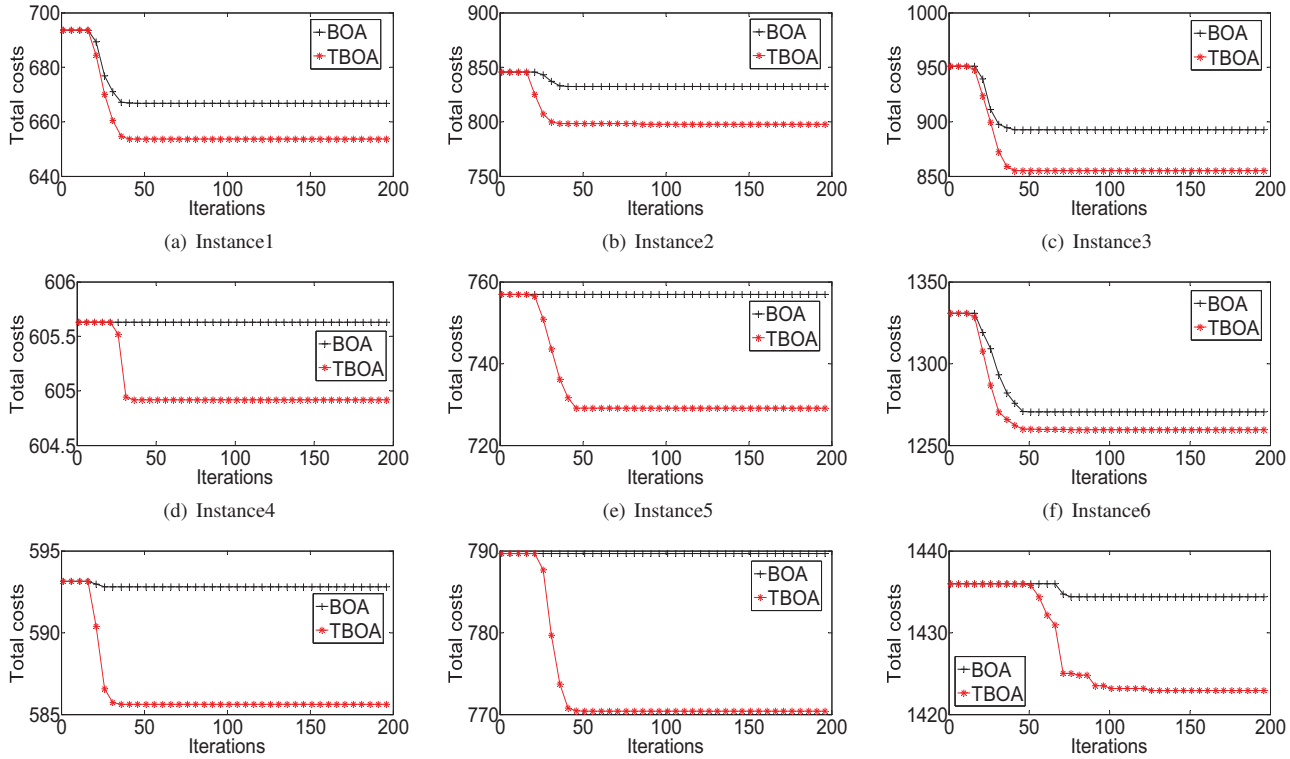
It is observed from Table I that TBOA outperforms BOA in terms of the average total costs and the average computational cost performance among all the instances. The evolution of the mean total costs derived from the two algorithms is displayed in Fig. 3. From these results, we can easily draw a conclusion that TBOA performs much better than BOA. Its convergence is faster than BOA's significantly without losing the accuracy. Especially in terms of efficiency, TBOA can reduce the average computational cost by more than 80%. These results indicate that TBOA is highly competitive at solving the task assignment problems. The superior performance of TBOA can be attributed in large part to the use of a team of BOAs. Each BOA can precisely search promising offspring in a dimension. There seems to be an interesting behavior of TBOA taking a global search in the early stages of runs since multiple Bayesian networks maintain the diversity. Then as iterations proceed, TBOA gradually converge. The conditional probabilities among different values also gradually converge, thereby equipping TBOA with better local search ability.

VII. CONCLUSIONS

In this paper, the contribution of our work is that we for the first time propose a team of Bayesian optimization algorithms (TBOA) to search and learn dimensionality. Each BOA undertakes a dimensional search. The results on a task assignment problem in heterogeneous computing systems show that the proposed TBOA can significantly decrease the computational cost of the overall training while maintaining high solution accuracy. In the future, we will upgrade the performance of TBOA through a coordination mechanism. Besides, we will try to apply TBOA to solve more discrete optimization problems, such as multiobjective resource-constrained project scheduling problems [4] and optimal power flow problems [15].

ACKNOWLEDGMENT

This work is supported by China NSF under Grants No. 61272271, 71371142, 61332008 and 91218301, Shanghai NSF under Grant No. 12ZR1434000, National Basic Research Program of China under Grant No. 2014CB340404, International Cooperation Project of Chinese Ministry of Science and Technology under Grant No. 2012DFG11580.



(a) Instance1 (b) Instance2 (c) Instance3

(d) Instance4 (e) Instance5 (f) Instance6

(g) Instance7 (h) Instance8 (i) Instance9

Fig. 3. The evolution of the mean total costs derived from TBOA and BOA over 20 independent runs on each of 9 instances.

REFERENCES

[1] H. Mühlenbein and G. Paaß,"From recombination of genes to the estimation of distributions I. Binary parameters," *in Parallel problem solving from nature, PPSN, 1996, IV. Lecture notes in computer science.* pp. 178-187, 1996.

[2] V. A. Shim, K. C. Tan and C. Y. Cheong,"A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem," *IEEE Trans. on Systems, Man, and Cybernetics-Part C: Applications and Reviews.* vol. 42, no. 5, pp. 682-691, 2012.

[3] J. H. Wang, Y. L. Zhou, J. Yin and Y. N. Zhang,"Competitive hopfield network combined with estimation of distribution for maximum diversity problems," *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics.* vol. 39, no. 4, pp. 1048-1066, 2009.

[4] L. Wang, C. Fang, C. D. Mu and M. Liu,"A pareto-archived estimation-of-distribution algorithm for multiobjective resource-constrained project scheduling problem," *IEEE Trans. on Engineering Management.* vol. 60, no. 3, pp. 617-626, 2013.

[5] M. Pelikan, D. Goldberg and E. Cantá-Paz, "BOA: the Bayesian optimization algorithm," *in Proc. of the 1999 Genetic and Evolutionary Computation Conference, 1999. (GECCO'99).* pp. 525-532, 1999.

[6] Y. Katsumata and T. Terano, "Bayesian optimization algorithm for multi-objective solutions: Application to electric equipment configuration problems in a power plant," *The Congress on Evolutionary Computation, 2003. (CEC'03).* vol. 2, pp. 1101-1107, 2003.

[7] J. Schwarz, J. Ocenasek and J. Jaros, "Advanced Bayesian optimization algorithms applied in decomposition problems," *in Proc. of the 11th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS'04).* pp. 102-111, 2004.

[8] X. X. Liu, J. P. Shi, W. G. Zhang and Y. Wu, "An improved Bayesian optimization algorithm for fault identification on flight control system," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2008.* pp. 825-828, 2008.

[9] J. P. Li and U. Aickelin, "A Bayesian optimization algorithm for the nurse scheduling problem," *The Congress on Evolutionary Computation. (CEC'03).* pp. 2149-2156, 2003.

[10] J. Yang, H. Xu, L. Pan, P. Jia, F. Long and M. Jie, "Task scheduling using Bayesian optimization algorithm for heterogeneous computing environments," *Applied Soft Computing.* vol. 11, no. 4, pp. 3297-3310, 2011.

[11] M. Pelikan, D. E. Goldberg and S. Tsutsui, "Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms," *SICE 2003 Annual Conference.* vol. 3, pp. 2738-2743, 2003.

[12] J. Ocenasek and J. Schwarz, "Estimation of distribution algorithm for mixed continuous-discrete optimization problems," *in Proc. of the 2nd Euro-International Symposium on Computational Intelligence.* pp. 227-232, 2002.

[13] C. W. Ahn, R. S. Ramakrishna and D. E. Goldberg, "Real-coded Bayesian optimization algorithm: Bringing the strength of BOA into the continuous world," *Genetic and Evolutionary Computation-GECCO.* pp. 840-851, 2004.

[14] M. T. Indrawan, B. Srinivasan, C. C. Wilson and R. Redpath, "Optimising Bayesian belief networks: A case study of information retrieval systems," *IEEE International Conference on Systems, Man, and Cybernetics.* vol. 3, pp. 525-532, 1998.

[15] Q. Kang, M. C. Zhou, J. An, and Q. Wu, "Swarm intelligence approaches to optimal power flow problem with distributed generator failures in power networks," *IEEE Trans. on Automation Science and Engineering.* vol. 10, no. 2, pp. 343-353, 2013.

[16] X. Liang, W. Li, Y. Zhang, and M. Zhou, "An adaptive particle swarm optimization method based on clustering," *Soft Computing, DOI 10.1007/s00500-014-1262-4, available on-line.* 2014.

[17] S. M. Shatz, J. P. Wang and M. Goto, "Task allocation for maximizing reliability of distributed computer systems," *IEEE Trans. on Computers.* vol. 41, no. 9, pp. 1156-1168, 1992.