Maintaining optimal state probabilities in biological systems

Madhumita Ghosh · Basant K. Tiwary ·
Dilip Datta

Received: 27 January 2010/Revised: 12 May 2010/Accepted: 13 May 2010/Published online: 23 June 2010
© Springer Science+Business Media B.V. 2010

Abstract A biological problem is usually studied experimentally by reducing it into a number of modules. In contrast, the systems biology approach seeks to address the collective behavior of interacting molecules vis-a-vis the corresponding higher level behavior. Various attributes of a biological system are conditionally dependent on each other, and these conditionalities are usually represented through Bayesian networks for computing easily the joint probability for a state of an attribute. In this article, a genetic algorithm is investigated to a biological system, by representing it through a Bayesian network, for evaluating the optimum state probabilities of different attributes, in order to obtain a desired joint probability for a given state of an attribute. We believe that such a study would be helpful in achieving a desired health condition by maintaining various attributes of a system to their estimated optimum levels.

Keywords Systems biology · Bayesian network · State probability · Joint probability · Genetic algorithm

Introduction

Traditionally, biology is studied experimentally by reducing a problem into a number of components or modules, which are then integrated on the basis of some assumptions for a reasonably good understanding of the problem. However, the advent of genome-wide high-throughput technologies on various facets of gene expression, function and evolution, has significantly influenced researchers to move from the reductionism to an integration using systems biology approach. The systems biology approach can be applied to explain different phenomena of a biological system, not on a gene-by-gene basis, but through the behavior and relationships among all the functioning elements of the system.

Various attributes of a biological system are interdependent in most of the cases, and their cyclic changes play critical roles in governing the system. The probability of occurrence of a state of one attribute is conditional on that of another attribute. The calculation of such conditionalities by the traditional probability theory is often NP-hard. This is because the biological attributes are usually discrete random variables that lie in undirected cycles when represented in terms of their dependency. Therefore, these relationships are usually represented through Bayesian networks for reducing the computational complexity, by exploiting individual state probabilities, in calculating a joint probability for an attribute (Heckerman 1996; Mitra and Acharya 2003; Needham et al. 2006; Sierra et al. 2004). Sometime a reverse computation may also be required, in which the state probabilities of an attribute are to be calculated in order to obtain a desired joint probability for a state of another attribute. Since various attributes of a biological system are dependent on each other, it is likely that, to maintain a particular condition for a given

M. Ghosh · B. K. Tiwary
Department of Life Sciences, Assam University, Silchar, Assam
788 011, India
e-mail: madhu_00713@yahoo.co.in

B. K. Tiwary
e-mail: basant68@email.com

D. Datta (✉)
Department of Mechanical Engineering, National Institute
of Technology, Silchar, Assam 788 010, India
e-mail: datta_dilip@rediffmail.com

attribute, not only that attribute but its many parent attributes are also to be treated. Therefore, such a reverse computation should be helpful in estimating the optimum levels of different attributes, to which the attributes are to be treated, in order to achieve a desired health condition. However, deterministically only one state probability can be calculated at a time from a single known joint probability, which motivates the use of a random tool for evaluating simultaneously more than one state probability from a single joint probability. For such a situation, a genetic algorithm is investigated here for evaluating the optimum state probabilities for different attributes, so that a desired joint probability can be obtained for a given state of another attribute. The probabilities, whose optimum values are to be evaluated, may not have any prior known value or their already known values may need to be rectified. The effectiveness of the proposed technique is presented here through its application to a case study under different conditions. To our best knowledge, this is the first work attempting to evaluate the optimum state probabilities in a network using an optimization tool.

The main contribution of this work includes the introduction of the novel concept of optimizing state probabilities of different attributes of a biological system, along with the presentation of the corresponding computation in detail, which should find an important application in the medical science in diagnosing and treating a disease by maintaining an attribute in the most suitable condition from alternative options.

The rest of the article is organized as follows: the Bayesian network, along with the detail computation of a joint probability, is introduced in Section “[Introduction to Bayesian network](#)”. A biological system, represented through a Bayesian network, is formulated as an optimization problem in Section “[Formulating a biological system as an optimization problem](#)”, followed by Section “[A genetic algorithm for optimizing a biological system](#)” addressing the proposed genetic algorithm for optimizing the state probabilities of the biological system. The numerical experiment of the proposed technique is presented in Section “[Numerical experiments](#)”. Finally, the article is concluded in Section “[Conclusions](#)” with a summary of the present work and the future scope on the theme of this article.

Introduction to Bayesian network

A Bayesian network is a directed acyclic graph showing the causal relationships among various attributes of a probability distribution, i.e., a Bayesian network can be defined as the graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V} = \{v_1, v_2, \dots, v_n\}$ denotes the set of n nodes of $\mathcal{G}$, and $\mathbb{E} =$

$\{e_{ij} | i, j = 1, 2, \dots, n; i \neq j\}$ denotes the set of edges connecting the nodes. Each node of $\mathcal{G}$ represents a discrete attribute, which has a finite number of states bearing some discrete probability values based on the occurrence of its parent attributes. The edge e_{ij} may be assigned the value of 1 if v_i and v_j are connected, meaning that v_j is conditionally dependent on v_i , otherwise $e_{ij} = 0$ reflecting that v_i and v_j are conditionally independent. $\mathcal{G}$ is directed in the sense that $e_{ij} = 1$ indicates that v_j is conditionally dependent on v_i , but v_i is independent of v_j . On the other hand, $\mathcal{G}$ is acyclic because it does not have any loop or feedback system.

A Bayesian network is used for probabilistic inference among various attributes of a conditional probability distribution, and it is one of the most widely used methods in exploring huge data, particularly noisy/uncertain data. Obtaining a Bayesian network from given data is a learning process that can be divided into two parts—the qualitative structural learning with the determination of the topology of the structure, and the quantitative parametric learning with the estimation of the probabilities of different states of the structure (Felgaer et al. 2006). Once a Bayesian network is constructed completely, it constitutes an efficient device to perform probabilistic inference among its attributes. In the general case of a Bayesian network, consisting of a set of n attributes $\mathbb{V} = \{v_1, v_2, \dots, v_n\}$ with the discrete probability values of $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$, the joint probability distribution can compactly be expressed as below:

$$p(x_1, x_2, \dots, x_n) = \prod_{i=1}^n p(x_i | \mathbf{pa}(x_i)), \quad (1)$$

where $\mathbf{pa}(x_i)$ represents all the parents of v_i , and the term $p(x_i | \mathbf{pa}(x_i))$ is just the prior probability of v_i in $\mathbf{pa}(x_i)$. Given the state probabilities of one or more attributes, the state of other posterior/prior attributes can be inferred using Eq. (1), which in the expanded form is the summation of the probabilities of all the routes in the Bayesian network. If the state of the prior attribute v_i is known to be a , the probability, that the state of the posterior attribute v_j ($j > i$) would be b , can be given by Eq. (2) as follows:

$$p(v_j = b | v_i = a; j > i) = \sum_{y_1} \dots \sum_{y_m} \left. \begin{aligned} & p(v_j = b | \mathbf{pa}(v_j = b)) \cdot \\ & p(v_{j-1} = y_1 | \mathbf{pa}(v_{j-1} = y_1)) \cdot \\ & \dots \dots \dots \cdot \\ & p(v_{i+1} = y_m | v_i = a) \end{aligned} \right\} \quad (2)$$

where $\mathbf{pa}(v_j = b)$ are the parent states of the attribute v_j for its given state b , m is the number of intermediate attributes in the route from v_j to v_i , and $(y_1, \dots, y_m)$ are the states of those intermediate attributes. On the other hand, if the state of the posterior attribute v_j is known to be b , the probability, that the state of the prior attribute v_i ($i < j$) would be a , can be given as below:

$$p(v_i = a | v_j = b; i < j) = \frac{p(v_j = b | v_i = a) p(v_i = a)}{\sum_{k=1}^{s_i} p(v_j = b | v_i = \alpha_k) p(v_i = \alpha_k)}; \quad (3)$$

where s_i is the number of states in v_i , and α_k is its k -th state. The expansion of $p(v_j = b | v_i = a)$ or $p(v_j = b | v_i = \alpha_k)$ in Eq. (3) is similar with that given by Eq. (2).

Formulating a biological system as an optimization problem

A number of studies are reported in the literature on learning the optimal structures of Bayesian networks (Gao et al. 2007; Guo et al. 2002; Habrant 1999; Hsu et al. 2002; Kabli et al. 2007; Larrañaga et al. 1996, 1997; Park and Cho 2006). However, no study could be found optimizing the state probabilities of biological systems or any other Bayesian network. In such an attempt, a biological system is formulated here as an optimization problem, in which the optimum state probabilities of different attributes of a Bayesian network are evaluated in order to obtain a desired joint probability for a given state of an attribute. The probabilities of the network, whose optimum values are to be evaluated, are taken as the variables of the problem. The objective function of the problem is to minimize the deviation of the evaluated joint probability from its desired value (the ideal deviation is zero).

The state probabilities of a biological system are subject to four types of constraints. For illustrating the constraints, a simple cell-signaling pathway is considered, which is borrowed from Needham et al. (2006) and shown in Fig. 1. The pathway consists of six attributes: a stimulant (ST), an extracellular signal (SI), an inhibitor of the signal (IN), a G protein-coupled receptor (RE), a G protein (GP), and the cellular response (CR). Out of these six attributes, the stimulant is the root, and all other attributes are dependent on each other. Moreover, the receptor has two parents: the inhibitor and the signal. With the help of this example, the four types of constraints involved with the state probabilities of a biological system are stated below:

1. A probability value lies in the range of [0,1].
2. The total of all the state probabilities of an attribute is one (the sum of probabilities in any row of a probability table in Fig. 1 is one).
3. If an attribute has more than two states, their probabilities should be in an order (the probabilities in any row of the inhibitor in Fig. 1 are in ascending/descending order). This requirement is obvious if the states are also in an order, e.g., the states of *high*, *medium* and *low* of the inhibitor in Fig. 1 are in an order and hence, their values also should be in an order.

4. The probabilities of any state of an attribute, corresponding to the states of its each parent attribute (against fixed states of other parent attributes), are also in an order (the probabilities in any column of the receptor in Fig. 1, corresponding to the states of each parent attribute against fixed states of other parent attributes, are in ascending/descending order). Like in constraints (3) above, this ordering is required if a parent attribute has more than two states, e.g., the receptor in Fig. 1, which has two parent attributes (signal and inhibitor) each having three states.

Under the above constraints, the optimization problem in hand can be expressed mathematically as given by Eq. (4) below:

$$\begin{aligned} & \text{Minimize} \quad |p_{\text{eval}} - p_{\text{des}}| \\ & \text{Subject to:} \quad (1) \quad 0.0 \leq x_i \leq 1.0; \quad i = 1 \text{ to } n \\ & \quad (2) \quad \sum_{m=1}^{s_j} p_{jklm} = 1.0; \quad j = 1 \text{ to } a; k = 1 \text{ to } s_j^{(1)}; l = 1 \text{ to } s_j^{(2)} \\ & \quad (3) \quad p_{jkl1} \leq \dots \leq p_{jkl s_j^{(1)}} \quad \text{or} \quad p_{jkl1} \geq \dots \geq p_{jkl s_j^{(1)}}; \quad j = 1 \text{ to } a; \\ & \quad \quad \quad k = 1 \text{ to } s_j^{(1)}; \\ & \quad \quad \quad l = 1 \text{ to } s_j^{(2)} \\ & \quad (4) \quad p_{jk1m} \leq \dots \leq p_{jk s_j^{(2)} m} \quad \text{or} \quad p_{jk1m} \geq \dots \geq p_{jk s_j^{(2)} m}; \quad j = 1 \text{ to } a; \\ & \quad \quad \quad k = 1 \text{ to } s_j^{(1)}; \\ & \quad \quad \quad m = 1 \text{ to } s_j; \\ & \quad \quad \quad p_{j1lm} \leq \dots \leq p_{j s_j^{(1)} lm} \quad \text{or} \quad p_{j1lm} \geq \dots \geq p_{j s_j^{(1)} lm}; \quad j = 1 \text{ to } a; \\ & \quad \quad \quad l = 1 \text{ to } s_j^{(2)}; \\ & \quad \quad \quad m = 1 \text{ to } s_j; \end{aligned} \quad (4)$$

where p_{eval} is the joint probability evaluated by an optimization technique for a given state, and p_{des} is its desired joint probability. x_i is the i -th state probability of an attribute which is to be optimized, and n is the total number of such state probabilities of that attribute. Constraints (2)–(4) in Eq. (4) are given for two parent attributes only, which can be altered as per the requirement of an attribute, e.g., the root attribute has no parent, and hence constraint (4) is not applicable to it. a is the number of attributes whose probabilities are to be optimized, s_j is the number of states of the j -th one of those attributes, and $s_j^{(1)}$ and $s_j^{(2)}$ are, respectively, the numbers of states of the first and second parents of that j -th attribute. Accordingly, the number of inequality sets under constraint (4), for a single attribute (say, the j -th attribute), becomes s_j if $d_j = 1$, $s_j \sum_{k=1}^{d_j} s_j^{(k)}$ if $d_j = 2$, $s_j \sum_{k=1}^{d_j-1} \sum_{l=1}^{d_j} s_j^{(k)} s_j^{(l)}$ if $d_j = 3$, and so on, where d_j is number of parents of the j -th attribute. Therefore, the number of ascending/descending inequality sets, under constraint (4), for a three-state attribute with a three-state parent (say, the inhibitor in Fig. 1) is only 3, while this number for a two-state attribute with two parents, each having three states (say, the receptor in Fig. 1), is 12. It is to be mentioned that some or all of the probabilities of constraints (2)–(4) are variables (x_i , $i = 1$ to n) given under constraint (1).

Fig. 1 Bayesian network for a simple cell-signaling pathway and its state probabilities (Needham et al. 2006)

Since many numerical methods usually suffer from the difficulty in handling equality constraints, a simple repairing mechanism is proposed here for working with the equality constraint (2) of Eq. (4). The mechanism works by equally distributing the excess/lacking amount of probabilities among the individual states of an attribute. Mathematically, the working procedure of the proposed repairing mechanism is given by Eq. (5),

$$p'_{jklm} = p_{jklm} + d; \quad \text{if } 0 \leq p_{jklm} + d \leq 1.0$$

$$\text{where, } d = \begin{cases} -\frac{1}{s_j} \sum_{m=1}^{s_j} p_{jklm}; & \text{if } \sum_{m=1}^{s_j} p_{jklm} > 1; \\ \frac{1}{s_j} \sum_{m=1}^{s_j} p_{jklm}; & \text{if } \sum_{m=1}^{s_j} p_{jklm} < 1; \\ 0; & \text{otherwise;} \end{cases} \quad (5)$$

where p'_{jklm} is the modified value of p_{jklm} , and d is the excess/lacking amount of probability that is to be added to or subtracted from each state probability. All other notations in Eq. (5) are the same with those in Eq. (4).

A genetic algorithm for optimizing a biological system

The genetic algorithm (GA) is a stochastic search technique that mimics the mechanisms of the Darwinian evolution based on the concept of the *survival of the fittest* (Deb 2001; Goldberg 1989). The very basic and most crucial component of a GA is the *solution representation*, popularly known as the *chromosome*, which represents a complete solution of a problem. A GA begins with a set of random solutions, referred to as a *population*, which is evolved over generations (iterations) by the repeated applications of some genetic operators analogous to ones from natural evolution, like selection, crossover, and mutation. A selection operator emphasizes good solutions

and eliminates weak solutions by forming a temporary population, known as the *mating pool*, with multiple copies of good solutions from the original population. A crossover operator generates children solutions by applying some probability distribution to the parent solutions of the mating pool. A mutation operator explores the neighborhood of a child solution generated by the crossover operator. In this way, the population is gradually improved towards the optimum value of a given objective function, also called the *fitness function*, which is a measure of the quality of a solution. The process of evolution of the population is continued until some termination criteria are met, usually until a predefined maximum number of generations are performed or the desired objective value is obtained.

As per the formulation presented in Section “Formulating a biological system as an optimization problem”, a GA is investigated here for evaluating the optimum state probabilities of different attributes of a Bayesian network, so that a desired joint probability can be obtained for a given state of an attribute. A real-coded chromosome is chosen for representing the real-valued variables (unknown state probabilities) of the problem, in which the length of a chromosome is equal to the number of variables and the value of an element of the chromosome indicates the value of the representing variable. A mating pool is formed by applying the *binary tournament selection operator* (Deb 2001), which picks up two random solutions (chromosomes) from the population and stores a copy of the best solution in the mating pool. The process is repeated until the size of the mating pool equals that of the population. Once the mating pool is complete, the *simulated binary crossover operator* (Deb 2001) is applied as given by Eq. (6) below, which takes two random parent solutions from the mating pool and generates two children solutions by crossing them with a predefined *crossover probability*:

$$\left. \begin{aligned} x_j^{(c_1)} &= \frac{1}{2} \left[(1 + \bar{\beta})x_j^{(p_1)} + (1 - \bar{\beta})x_j^{(p_2)} \right]; \quad j = 1, 2, \dots, l; \\ \text{and } x_j^{(c_2)} &= \frac{1}{2} \left[(1 - \bar{\beta})x_j^{(p_1)} + (1 + \bar{\beta})x_j^{(p_2)} \right]; \quad j = 1, 2, \dots, l; \end{aligned} \right\} \quad (6)$$

where x_j is the j -th variable of a chromosome and l is the length of the chromosome. p and c denote parents and children, respectively. $\bar{\beta}$ is the ordinate of a probability distribution, which is chosen in such a way that the area under the probability curve from 0 to $\bar{\beta}$ equals a random number u ($0 < u < 1$), so that it can be expressed as below:

$$\bar{\beta} = \begin{cases} (2u)^{\frac{1}{1+n}}, & \text{if } u \leq 0.5 \\ \left(\frac{1}{2(1-u)} \right)^{\frac{1}{1+n}}, & \text{otherwise;} \end{cases} \quad (7)$$

where n is a non-negative probability distribution index. The process of generating children solutions is also repeated until the size of the children population equals that of the original population. After generating a children population, the *polynomial mutation operator* (Deb 2001) is applied, with a small predefined *mutation probability*, for exploring the neighborhood of a child solution, in which, for a random number r in (0,1) and a given polynomial distribution index $\eta > 0$, the variable x_j is evolved in the range of $(x_j^{(l)}, x_j^{(u)})$ as given below:

$$x_j \leftarrow x_j + (x_j^{(u)} - x_j^{(l)})d_q \quad (8)$$

where

$$d_q = \begin{cases} \left[\left(2r + (1 - 2r) \left(1 - \frac{x_j - x_j^{(l)}}{x_j^{(u)} - x_j^{(l)}} \right)^{\eta+1} \right)^{\frac{1}{\eta+1}} - 1, & \text{if } r < 0.5 \\ 1 - \left[2(1 - r) + 2(r - 0.5) \left(1 - \frac{x_j^{(u)} - x_j}{x_j^{(u)} - x_j^{(l)}} \right)^{\eta+1} \right]^{\frac{1}{\eta+1}}, & \text{otherwise.} \end{cases}$$

For retaining the best solutions of a generation, after the completion of the mutation operation, both the original population and the mutated children population are combined. Then the combined solutions are sorted in terms of their objective values, and finally the best 50% solutions are extracted as the initial population for the next generation. Further, the *penalty-parameter-less constraint handling approach*, proposed by Deb (2000), is applied for working with infeasible solutions. In this approach, all the infeasible solutions are first made inferior to any feasible

solution through a fitness function, and then all the solutions are treated as feasible solutions only.

Numerical experiments

A computer program is developed in C programming language, implementing Eqs. (2) and (3) for computing joint probabilities in a Bayesian network, and Eqs. (4) and (5) for optimizing state probabilities using the proposed GA. The effectiveness of the proposed technique is demonstrated here by applying the developed program to the Bayesian network of Fig. 1 for the following two randomly chosen cases, each of which is considered under two separate conditions:

1. Optimizing the state probabilities of the inhibitor with the desired joint probability of
 - (a) 0.59 for the G protein to be active if the stimulant is present,
 - (b) 0.44 for the stimulant to be present if the G protein is active,
2. Optimizing the state probabilities of the receptor with the desired joint probability of
 - (a) 0.59 for the G protein to be active if the stimulant is present, and
 - (b) 0.44 for the stimulant to be present if the G protein is active.

The desired joint probabilities in the above cases are calculated by applying Eqs. (2) and (3) to the state probabilities shown in Fig. 1. However, in order to obtain the desired joint probabilities through the proposed GA, the related state probabilities are considered to be unknown, i.e., the state probabilities of the inhibitor in the case (1) and those of the receptor in the case (2) are made unknown. Under these considerations, the number of variables (number of unknown state probabilities) and number of constraints (as per the formulation given in

Eq. (4)) in each condition of the case (1) become 9 and 27, respectively. Similarly, those in each condition of the case (2) are 18 and 66, respectively.

Since the performance of a randomized technique may be influenced by initial solutions as well as its other user-defined parameter values (Datta 2007; Reyes-Sierra and Coello 2006), the proposed GA is applied 30 times to each of the above four conditions with different initial solutions and other parameter settings, which are summarized in Table 1.

The number of solutions in the population of each run is randomly fixed in the range of [50, 100], where a solution is a real-valued array of variables. A population is initialized by random real numbers in the range of (0, 1), which is then improved over generations towards the optimum objective value. The maximum number of generations allowed to perform is 1000, along with an additional termination criterion that the execution would be terminated if the desired joint probability is obtained, within a predefined accuracy (two-decimal places of accuracy is considered in the present study), at least in one solution of the population. The crossover probability is varied, over 30 runs of each condition, from 61 to 90% at an interval of 1%. On the other hand, instead of assigning a fixed user-defined value, the mutation probability is made self-adaptive in the range of (0, 5%), for which Eq. (8) is used to evolve it over generations in the same way as a real variable is mutated. The mutation probability is made self-adaptive as it is already learned that the performance of an algorithm on a combinatorial problem becomes highly dependent on its user-defined parameters, which can be reduced by making some parameters self-adaptive over generations (Datta 2007; Reyes-Sierra and Coello 2006).

The developed program is executed in the Fedora 8 Linux environment in a HCL Desktop with 2.67 GHz processor, 2.0 GB RAM and 160 GB HDD. The best results in terms of number of function evaluations, obtained by applying the proposed technique to the considered four conditions, are given in Tables 2, 3, 4 and 5, respectively. Each Table includes the optimized state probabilities, size of the population, number of generations performed until the desired joint probability is obtained, number of function evaluations in obtaining that desired joint probability, and the total computational time.

Table 1 GA related parameter values used in each of the four numerical experiments

Parameter	Value
Number of runs	30
Number of solutions in the population	[50, 100]
Maximum number of generations to be performed	1000
Crossover probability	61–90% at an interval of 1%
Distribution index for the crossover operator	2
Mutation probability	Self-adaptive in (0, 5%)
Distribution index for the mutation operator	30

Moreover, the overall computational costs, in terms of computational time and number of function evaluations, for the considered four conditions are summarized in Table 6, which gives the range of computational time over 30 runs of each condition, range of function evaluations, and the mean, median and standard deviation of the numbers of function evaluations.

It is observed in Tables 2, 3, 4 and 5 that the proposed technique is successful in obtaining the desired joint probabilities, within the predefined accuracy (two decimal places), under all of the considered four conditions. Table 6 shows that the computational cost is also very nominal in obtaining such results (of course, the computational cost would increase with increasing accuracy).

Table 2 Optimum state probabilities of the inhibitor for $p(\text{GP} = \text{Active} \mid \text{ST} = \text{Present}) = 0.59$

IN p (inhibitor/signal)			
Signal	High	Medium	Low
High	0.41	0.34	0.25
Medium	0.40	0.32	0.28
Low	0.21	0.26	0.53

Size of the population (pop_size) = 52

Number of generations performed (gen) = 23

Number of function evaluations ($= pop_size \times gen$) = 1196

Computational time < 1 s

Table 3 Optimum state probabilities of the inhibitor for $p(\text{ST} = \text{Present} \mid \text{GP} = \text{Active}) = 0.44$

IN p (inhibitor/signal)			
Signal	High	Medium	Low
High	0.48	0.35	0.17
Medium	0.30	0.30	0.40
Low	0.26	0.28	0.46

Size of the population (pop_size) = 56

Number of generations performed (gen) = 44

Number of function evaluations ($= pop_size \times gen$) = 2464

Computational time < 1 s

Table 4 Optimum state probabilities of the receptor for $p(\text{GP} = \text{Active} \mid \text{ST} = \text{Present}) = 0.59$

RE p (receptor/signal,inhibitor)			
Signal	Inhibitor	Yes	No
High	High	0.65	0.35
High	Medium	0.65	0.35
High	Low	0.41	0.59
Medium	High	0.33	0.67
Medium	Medium	0.65	0.35
Medium	Low	0.66	0.34
Low	High	0.24	0.76
Low	Medium	0.72	0.28
Low	Low	0.73	0.27

Size of the population (pop_size) = 50Number of generations performed (gen) = 141Number of function evaluations ($= pop_size \times gen$) = 7050

Computational time < 1 s

Table 5 Optimum state probabilities of the receptor for $p(\text{ST} = \text{Present} \mid \text{GP} = \text{Active}) = 0.44$

RE p (receptor/signal,inhibitor)			
Signal	Inhibitor	Yes	No
High	High	0.29	0.71
High	Medium	0.67	0.33
High	Low	0.93	0.07
Medium	High	0.63	0.37
Medium	Medium	0.60	0.40
Medium	Low	0.60	0.40
Low	High	0.74	0.26
Low	Medium	0.40	0.60
Low	Low	0.25	0.75

Size of the population (pop_size) = 64Number of generations performed (gen) = 80Number of function evaluations ($= pop_size \times gen$) = 5120

Computational time = 1 s

Table 6 Computational costs over 30 runs of each of the four numerical experiments

Case	Time (s)	Function evaluations			
		Range	Mean	Median	St. Deviation
1(a)	0–1	1196–9968	6107.733	6105	52.729
1(b)	0–1	2464–9828	6635.867	6890	49.641
2(a)	0–1	7050–28890	17365.533	15516	76.927
2(b)	0–1	5120–25164	13665.867	13583	70.895

The most important observation made by comparing the state probabilities in Fig. 1 and Tables 2, 3, 4 and 5 is that the GA can result in different sets of state probabilities for an attribute against the same joint probability of a given state of another attribute. The physical significance of this type of results in a system of attributes is that a parent attribute does not need to have any fixed condition in order to maintain a particular condition for a dependent attribute, but it may remain in different conditions also. Such outcome might be a very interesting finding to the medical science in diagnosing and treating a disease by maintaining an attribute in the most suitable condition from alternative options. The feasibility and acceptability of such results in the case of a real biological system may be validated experimentally (the experimental part is not covered in this study). If needed, more restrictive results can also be produced by imposing additional constraints. For example, the presence of a particular state probability higher than 12% may not be acceptable for maintaining a desired health condition, in which case its range of variation would be [0,0.12] instead of [0, 1].

Conclusions

A novel concept of optimizing state probabilities of different attributes of a biological system is introduced. A genetic algorithm is investigated to a biological system, by representing it as a Bayesian network, for evaluating the optimum state probabilities of different attributes, so that a desired joint probability can be obtained for a given state of an attribute. Such an attempt to evaluate the optimum state probabilities of a biological system should be helpful in achieving a desired health condition by treating various attributes of a system to their estimated optimum levels. It is also observed from the numerical experiment that an optimization tool, such as a genetic algorithm, has the capability to produce different optimal conditions for a parent attribute in order to maintain a particular condition for a dependent attribute, which might be a very interesting outcome to the medical science in diagnosing and treating a disease by maintaining an attribute in the most suitable condition from alternative options. However, the feasibility and acceptability of such outcome in a real biological system needs to be validated experimentally. The future aim of this research is to conduct a comparative study by applying the proposed technique in some real biological systems, in hormone level, under normal and different diseased conditions.

Acknowledgments Authors are thankful to the unknown reviewers and the Editorial Office of the Journal for their valuable comments and suggestions in improving the article from its original version.

References

- Datta D (2007) Multi-objective evolutionary algorithms for resource allocation problems. PhD thesis, Department of Mechanical Engineering, Indian Institute of Technology Kanpur (IIT-Kanpur), India
- Deb K (2000) An efficient constraint handling method for genetic algorithms. *Comp Methods Appl Mech Eng* 186:311–338
- Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, Chichester
- Felgaer P, Britos P, Martínez R (2006) Prediction in health domain using Bayesian networks optimization based on induction learning techniques. *Int J Mod Phys* 17(3):447–455
- Gao S, Xiao Q, Pan Q, Li Q (2007) Learning dynamic Bayesian networks structure based on Bayesian optimization algorithm. In: Liu D et al (ed) *ISNN 2007, Part II*, vol 4492 of *Lecture Notes in Computer Science (LNCS)*, Springer, pp 424–431
- Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Boston
- Guo H, Perry BB, Stilson JA, Hsu WH (2002) A genetic algorithm for tuning variable orderings in Bayesian network structure learning (Abstract). In: *Proceedings of AAAI-2002*, American Association for Artificial Intelligence, 2002, pp 951–952
- Habrant J (1999) Structure learning of Bayesian networks from databases by genetic algorithms-Application to Time Series Prediction in Finance. In: *ICEIS*, Setubal, Portugal, pp 225–231
- Heckerman D (1996) A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research: Advanced Technology Division, Microsoft Corporation
- Hsu WH, Guo H, Perry BB, Stilson JA (2002) A permutation genetic algorithm for variable ordering in learning Bayesian networks from data. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, pp 383–390, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc
- Kabli R, Herrmann F, McCall J (2007) A chain-model genetic algorithm for Bayesian network structure learning. In: *Genetic and Evolutionary Computation Conference (GECCO-2007)*, pp 1264–1271, London, England
- Larrañaga P, Poza M, Yurramendi Y, Murga RH, Kuijpers CMH (1996) Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters. *IEEE Trans Pattern Anal Mach Intell* 18(9):912–926
- Larrañaga P, Sierra B, Gallego MJ, Michelena MJ, Picaza JM (1997) Learning Bayesian networks by genetic algorithms: a case study in the prediction of survival in malignant skin melanoma. In: Keravnou et al (ed) *Proceedings of the Artificial Intelligence in Medicine Europe (AIME-1997)*, vol 1211 of *Lecture Notes in Artificial Intelligence (LNAI)*, Springer, London, UK, pp 261–272
- Mitra S, Acharya T (2003) Data mining: multimedia, soft computing, and bioinformatics. Wiley, Chichester
- Needham CJ, Bradford JR, Bulpitt AJ, Westhead DR (2006) Inference in Bayesian networks. *Nat Biotechnol* 24(1):51–53
- Park H-S, Cho S-B (2006) An efficient attribute ordering optimization in Bayesian networks for prognostic modeling of the metabolic syndrome. In: Huang D-S et al (eds) *ICIC 2006*, vol 4115 of *LNBI*, Springer, pp 381–391
- Reyes-Sierra M, Coello CAC (2006) Multi-objective particle swarm optimizers: a survey of the state-of-the-art. *Int J Comput Intell Res* 2(3):287–308
- Sierra B, Lazkano E, Martínez-Otzeta JM, Astigarraga A (2004) Combining Bayesian networks, k nearest neighbours algorithm and attribute selection for gene expression data analysis. In: Webb, Yu (eds) *AI 2004*, vol 3339 of *LNAI*, Springer, pp 86–97