Bayesian hierarchical multi-objective optimization for vehicle parking route discovery

Romit S. Beed · Sunita Sarkar · Arindam Roy

Received: 15 March 2020 / Accepted: 21 August 2020 / Published online: 15 September 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract
Discovering an optimal route to the most feasible parking lot has been a matter of apprehension for any driver. Selecting the most optimal route to the most favorable lot aggravates further during peak hours of the day and at congested places. This leads to a considerable wastage of resources specifically time and fuel. This work proposes a Bayesian hierarchical technique for obtaining this optimal route. The route selection is based on conflicting objectives, and hence, the problem belongs to the domain of multi-objective optimization. A probabilistic data-driven method has been used to overcome the inherent problem of weight selection in the popular weighted sum technique. The weights of these conflicting objectives have been refined using a Bayesian hierarchical model based on multinomial and Dirichlet prior. Genetic algorithm has been used to obtain optimal solutions. Simulated data have been used to obtain routes which are in close agreement with real-life situations. Statistical analyses have shown the superiority of the weights obtained using the proposed algorithm based on Bayesian technique over the existing frequentist technique.

Keywords Bayesian · Genetic algorithm · Multi-objective · Optimization · Weighted sum technique

1 Introduction
Vehicle routing problems (VRPs) are popular combinatorial optimization problems present in transport sector which generally include allotting resources and scheduling in constrained environments from a depot to different locations having considerable financial implications. VRPs have gained popularity in the recent past because of their diverse applicability and commercial importance in defining effective distribution tactics to shrink operative costs in distribution networks. A typical VRP involves designing minimum cost routes from a principal warehouse to a set of physically distributed locations having varying requirements. Each such location needs to be serviced precisely once by a single van, and each van has a fixed carrying capacity. Vehicle parking route determination may be considered as an extension of a typical VRP. Instead of considering only distance as a parameter for determining the most appropriate parking lot, the time it takes to reach the particular lot and the availability of empty space may also be conflicting factors while determining the most suitable lot. Hence, this problem may be treated as a multi-objective optimization problem.

A multi-objective optimization problem (MOP) deals with two or more objectives or parameters contributing toward the final output, and in general, these objectives influence one another in a complicated and nonlinear manner [1]. The main goal is to obtain a set of values for these objectives, thereby generating an optimization of the overall problem. There has been extensive use of evolutionary computation in solving MOPs [2–5] as they provide high-quality solution through proper mapping of genetic information and fitness calculation. According to Goldberg, multi-objective or multi-criteria optimization is the process of optimizing multiple conflicting objectives in parallel subject to a set of constraints. In such problems, it is seen that there is not an individual solution...
that concurrently minimizes each objective to the fullest, but to a limit exceeding which the additional objective(s) will be compromised as a result [6]. After obtaining a particular solution, one of the main aims of MOPs is to compare it with other solutions and assess how improved this solution is in comparison with the existing set of solutions [7]. A multi-objective problem including several, conflicting objectives may be formulated into a one-objective scalar function. This popular a-priori technique known as weighted sum method or single-objective evolutionary algorithm (SOEA) is based on ‘linear aggregation of functions’ principle [8]. By definition, the weighted sum method reduces to a positively weighted convex sum of the objectives, as follows:

\[
\min \sum_{i=1}^{n} w_i \cdot f_i(x), \text{ where } \sum_{i=1}^{n} w_i = 1; \quad w_i > 0 \quad \forall i \quad (1)
\]

Minimization of this single-objective function is expected to give an efficient solution for the original multi-objective problem. The process involves scalarizing the conflicting objectives into a single-objective function. There are various scalarization techniques which have been proposed in the past. Zadeh popularized the weighted sum technique as a classical approach for solving such problems [9]. This method, as the name suggests, scalarizes a set of conflicting objective functions, by pre-multiplying each of the objective function by predefined weights. Though the technique is straightforward and computationally efficient, it may not be able to investigate all solutions when the true Pareto front in non-convex [10]. Multinomial distribution, being the multivariate generalization of binomial distribution, is used to model the probability of more than two quantitatively specified mutually exclusive outcomes occurring in a set of repeated trials and hence is a discrete multivariate distribution. Dirichlet distribution is multivariate generalization of beta, and it is well known that Dirichlet distribution acts as a conjugate prior for multinomial distribution.

Accounting for uncertainty is critical when there are various restrictions on data that may lead to erroneous analysis of choices and understandings. This is best done using the Bayesian analysis which is based on the Bayes’ theorem. Bayes models make it easier for researchers by reducing computational constraints through development of better and effective models of customer performance and decision making by incorporating a hierarchical model with Bayesian estimates. The sub-models are orderly integrated using the Bayes theorem for handling the uncertainty factor.

### 1.1 Objective and motivation

Majority of the existing works have aimed at choosing the set of weights in a manner such that it stabilizes the solution set [11–13]. The main objective of this work is to propose and design a model which will produce a much stable set of weights based on sound statistical concepts in comparison with that obtained deterministically. The criticisms of the existing methodologies for determination of weights have motivated this work and to propose the Bayesian model based on multinomial and Dirichlet priors. As per the authors’ existing knowledge, this work is first of its kind since none of the earlier works had this motivation of searching for stability in weights.

The algorithm has been so developed as to reflect the relative importance of the conflicting objectives through the respective weights, which were stochastically estimated, based on the data obtained from a pilot survey for the given purpose. Unlike the available continuation techniques, this method can be applied with convenience to handle any number of objectives. As this method yields a posterior probability distribution over the weights, the stochastically generated weight vectors can be used to obtain solutions with less computational complications.

Though a lot of research has been done in the field of vehicle routing problem, application of improved weights based on Bayes theorem to perform multi-objective route optimization is totally novel, and hence, this work is unique in this domain. The focus of this work has been to generate and use improved weights instead of focusing on improving the final result. The main objective of this work is to develop a flexible hierarchical Bayesian approach that adequately addresses various levels of uncertainties in the survey data pertaining to the vehicle routing problems that arise in the transport sector. In a nutshell, the novelty of the proposed methodology lies in the fact that the authors have used the Bayesian approach for the probabilistic modeling of the uncertainties associated with the weights which none of the existing methods have attempted earlier.

### 2 Literature review

The vehicle routing problem dates back to 1950s. Dantzig and Ramser [14] aimed to design a solution for delivering gasoline to petrol pump using a mathematical programming model. Ever since then, VRP has gained popularity and evolved substantially and is a subject of research in varied domains. Obtaining the most optimal route, thereby reducing the total transportation expense, is the main objective of this approach. However, in reality the modeling is much more complicated than the traditional VRP [15]. Thus, this traditional VRP model has evolved over time, constraints have been added like vehicle capacity, and the model has been named the capacitated vehicle routing problem (CVRP); another constraint is the time interval in which each customer has to be served named as the vehicle routing problem with Time Windows (VRPTW). Other variations of VRP which
exist are multiple depot VRP, periodic VRP, split delivery VRP, stochastic VRP, VRP with backhauls, VRP with pickup and delivering to name a few. Vehicle routing problem with Time Windows (VRPTW) is an improvised VRP where a time window is defined for each user [16]. Apart from the vehicle size limitation, each user defines a time frame within which an individual service has to be accomplished, e.g., packing or unpacking of items, parking a vehicle, etc. It may be possible that a particular vehicle may arrive before the scheduled time; in that case, it will have to wait till the scheduled start of service time. Han et al. [17] suggested the use of Wi-Fi positioning technology to resolve the parking problem by designing a technique of programmed assignment based on the personal demands of the users and avoidance of traffic conflicts. The major decision factors which were taken into consideration were lane usage conditions, driving distance, walking distance and the occupancy rate of parking space. Optimal routes were obtained using the Dijkstra’s algorithm; this route was then conveyed to the driver using the mobile network. Compared to traditional GPS positioning and mobile cellular network locating, Wi-Fi positioning technology provided better results thus leading to more effective use of the limited parking resources.

The Siemens Integrated Smart Parking Solution [18] is a traffic coordination system that streamlines driver decision making using heuristic data about parking space obtainability. Drivers are directed to the assigned parking location through the most optimum route using the in-car navigation system. RFID technique is used along with sensors to monitor, track and store data of every car involved in the system. The key features of the system are (i) dynamic guidance which helps reduce manual search and thereby lessens the traffic; (ii) improve route selection from source to destination using statistical data based on multiple objectives; (iii) gather knowledge and make available parking price and regulations as well as enforce parking violations fines cost-effectively by accessing real-time data; (iv) improve resource utilization by dynamically rerouting vehicles to optimize the usage of the parking lots at different times of the city. Xiao et al. [19] proposed an exact dynamic programming procedure for optimizing route schedules of dissimilar vehicles functioning in a time-varying traffic setting. The proposed algorithm demonstrated worst-case complexity of $O(nT^2)$ and showed that the time-dependent VRPs would no longer be more complex than VRPs. Lee et al. [20] proposed an innovative collision-free shortest path technique for mobile agents based on genetic algorithm (GA) and direction element for avoiding obstacles. The obstacle-based GA decreases the search spaces in comparison with free space-based techniques. Convergence occurs faster, and the algorithm requires less time to execute.

Genetic algorithm (GA) [6, 21] is a metaheuristic based on the concept of natural selection, designed to obtain, produce or choose a heuristic that provides a sufficiently favorable solution to an optimization problem, particularly with inadequate or partial information or restricted computing capacity. GAs fall under the category of evolutionary algorithms (EAs) and are used to obtain superior results to optimization and search requirements by basing on bio-inspired mechanisms like selection, crossover and mutation. It aims at generating the most optimal result to a problem within a vast set of likely possible results through genetic enhancement across generations. Holland [22] and his team at the University of Michigan presented an algorithm that modified a population of individual entities, each having a corresponding fitness value, into an evolved generation based on the Darwinian principle of evolution and policy of ‘survival of the fittest.’ Bryant et al. [23] suggested that genetic algorithms may be used in optimizing systems based on the natural phenomenon of evolution. The concept of survival of the fittest was incorporated in designing an algorithm which discovered or explored, though not all possible results, and obtained a fair result. The algorithm initiated by selecting random chromosomes from a pool of chromosomes and gradually generated the fittest solutions through several generations of iterations whereby the result of each generation was better than the result of the earlier generation.

GAs generate a population of solutions using the popular crossover and mutation operators. Chand et al. [24] proposed new methods for genetic operators—Sub Route Mapped Crossover Method (SMCM) and Sub Route Exchange Mutation Method (SEMM). They used dominant rank method to obtain Pareto optimal set, and the two objectives that they considered were number of vehicles and total cost (distance). The dominant rank method found optimum solutions efficiently. As GAs provide ease of operation and possess widespread applicability, it is one of the most trusted algorithms in computational optimization and operations research. A noble optimization technique requires balancing the degree of exploration of information gathered till the present generation through the processes of recombination and mutation with the degree of exploitation through the selection technique [25]. If the results gathered are exploited exceedingly, early convergence is evident. Conversely, if the search is extensive, it shows that the information gathered till then has not been used properly. By changing the parameters of the GA operators, exploration and exploitation can be controlled. Thus, genetic algorithms have become the most extensively used tool to solve the vehicle routing problem.

Selection of accurate weights can lead to better performance of an algorithm. Athan [26] proposed a quasi-random weighted criteria system that produces weights covering the Pareto set consistently. The method is based on random probability distribution and involves a large number of computations. Gennert and Yuille [27] proposed a nonlinear weight determination algorithm where an optimal
point is obtained that is not in the vicinity of the extreme points. Although a lot of work is available in the literature regarding systematic selection of the weights in solving a multi-objective optimization problem, till date a comprehensive data-driven technique determining weights reflecting the relative importance of the conflicting objectives is lacking. Most of the work in the available literature has focused on fixing the weights based on some prior beliefs or information. In some of the existing literature, the weights have been estimated simply by the proportion of preference in the respective categories. Pertaining to the above-mentioned concerns regarding selection of weights, this work has been based on the weights generated according to a hierarchical Bayesian technique which generates the weights based on data derived from a pilot survey.

Multi-objective optimization problems have recently been solved efficiently using metaheuristic techniques. Estimation of distribution techniques have been used extensively to effectively apply probabilistic modeling in this domain. Martins et al. [28] proposed a hybrid multi-objective Bayesian estimation of distribution algorithm for multiple objective states by focusing on the combined probability of decision variables, objectives and configuration constraints. When compared with other evolutionary algorithms, the proposed algorithm outperformed the existing approaches and majority of solutions generated very in the neighborhood of the Pareto front. Bayesian inference has developed into a very popular mechanism for enumerating the uncertainties in real-life problems. The Bayesian technique offers superior solutions by specifying appropriate prior distributions. Tang et al. [29] applied a Bayesian framework to an eco-hydrological model involving conflicting objectives. Outcomes indicate more consistent results using the proposed model highlighting the significance of proper prior distributions in such methods. Wada et al. [30] proposed an innovative non-heuristic Bayesian optimization technique that concurrently implemented multi-objective optimization as well as multi-point explorations. The authors proposed an approximation process for computing the gradient of the acquisition function. The efficiency of the technique was confirmed from the results of the various test functions.

Optimization of proposals and procedures institutes a pervasive problem in science and engineering. There are suitable optimization techniques that generate favourable results for intricate problems [31]. However, it is seen that when the optimization function is too complex and costly, majority of these techniques perform below par as they rely on a large set of executions of the objective functions [32, 33]. The Bayesian technique was a preferred solution to this set of continuous real-valued expensive functions [34]. Based on the Bayes theorem, the technique offered certain advantages: (i) they were exceedingly effective regarding the quantity of objective function assessments; (ii) they did not need any analytical understanding of the objectives [35]. Tajbakhsh [36] emphasized that Bayesian techniques work fine for multi-modal or non-convex objective functions. Galuzio et al. developed software named MOBOpt [37] for implementing multi-objective Bayesian optimization algorithm. The software was capable of generating the Pareto front approximation for various optimization problems. It required less evaluation of the objective functions in comparison to other techniques thus making it suitable for expensive objectives. The software produced favourable results when it was comprehensively tested on benchmark functions with up to 20 objective functions evaluations.

3 Multi-objective optimization model

Multi-objective optimization is becoming widespread in the current technology-driven optimization scenarios. Typical examples of multiple objectives are reduction in cost, increase in performance, growth of reliability, reduction in time, shortening of travel distance, maximization of speed, etc. It can be observed that many of these objectives may be conflicting with one another, and optimizing the problem with respect to one objective may generate undesired results with respect to the other objectives. A better approach to such a multi-objective problem is to discover a number of solutions, each of which satisfies the objectives up to a particular permissible level without being dominated by any other result. The two most common approaches for solving multiple-objective optimization are the utility theory method or the weighted sum method and the Pareto optimal method.

The weighted sum approach combines the different objective functions into a single function and then obtains an optimal solution. The drawback that remains with this approach is in deciding the weights of the different objectives. The Pareto optimal approach, on the other hand, obtains a set of values that are non-dominated with respect to each other. The weighted sum technique is generally employed to transform multiple objectives into one objective. This is implemented by issuing different weights to the various objectives which are chosen based on the objective’s relative significance with respect to the other objectives. The single-objective optimization aims at minimizing the overall cost as follows:

$$\min \ z = w_1 z_1(x) + w_2 z_2(x) + \cdots + w_k z_k(x),$$

where $z_i(x)$ are the normalized objective functions and $\sum w_i = 1$.

Developing a model to yield an optimal route to the lot involves multiple conflicting objectives. Here, we have explored the weighted sum technique to scalarize these objectives into a single-objective function. However, the weighted sum method suffers from the problem of scientific determi-
nation of weights for the conflicting objectives. This work is based on the weights generated by a novel Bayesian hierarchical method using Dirichlet distribution. The choice of weights is guided by a probability model which helps us draw data-based inference. To decide upon the weights, reflecting the relative importance of the objectives, a pilot survey is conducted. The number of individuals opting for the various mutually exclusive categories of conflicting objectives is noted as \( n_1, n_2 \) and \( n_3 \). Assuming \((n_1, n_2, n_3) \sim \text{Multinomial distribution with parameters} \, (n; w_1, w_2, w_3)\), where \( w_i \) (weight corresponding to the \( i \)th objective), is the proportion of individuals in the population (unknown) who will vote for category \( i \) of the objective function or is the probability that an individual selected randomly from the population votes for \( i \)th category. Assuming \((w_1, w_2, w_3)\) to be distributed as Dirichlet with concentration parameters \( a_1, a_2 \) and \( a_3 \), the estimates of weights are taken as

\[
w_i = \frac{a_i + n_i}{\sum_{i=1}^{3}(a_i + n_i)}, \quad i = 1, 2, 3 \tag{3}
\]

where \( a_1, a_2 \) and \( a_3 \) are the values maximizing the marginal likelihood function of data given \( a_1, a_2 \) and \( a_3 \).

The flowchart of the underlying architecture is as follows (Fig. 1).

### 3.1 Fitness function

This paper presents an innovative fitness function that models the real-life situation in the most appropriate way. One part of a city map has been modeled as shown in Fig. 2. Two locations ‘0’ and ‘1’ (marked in red) are considered to be the start locations. There are 26 intermediate junctions, ‘2’–‘27’ (marked in blue), and there are three parking lots, ‘28’–‘30’ (marked in green). The optimality and suitability of the parking lot as well as the path leading to it from the source is obtained using a fitness function that is formulated based on three factors—distance, speed and parking availability. Google Map API has been used to obtain the distance between these points as well as the average speed of journey in the same route.

The formulation of the minimization function of three conflicting objectives is as follows:

\[
\min \, f = \min(f_1, f_2, f_3) \tag{4}
\]

The first objective function minimization of distance function \( f_1 \) is formulated as follows:

\[
f_1 = \sum_{i,j \in N} d_{ij} \tag{5}
\]

where \( N \) = set of all nodes and \( d_{ij} \) is the distance travelled from node \( i \) to \( j \). \( f_1 \) summarizes the total distance travelled from the start node to the parking lot. The problem aims to minimize this distance \( f_1 \) in the main fitness function expression—the lesser the distance, the lesser the cost to reach the destination and the lower the overall fitness.
The second objective function is maximization of speed $f_2$ is defined as follows:

$$f_2 = -\frac{1}{n-1} \sum_{i,j \in N} s_{ij}$$

(6)

where $s_{ij}$ is the speed from node $i$ to node $j$, which may vary over different segments of the path. It is known that maximization $(f) \equiv$ minimization $(-f)$. The problem considers the average speed at which the car travels to the parking lot to be another factor which influences the fitness function—the greater the average speed, faster it reaches the destination, hence leading to minimization of the overall fitness function cost.

The third objective function is maximization of the parking availability and is defined as follows:

$$f_3 = -x_p$$

(7)

where $x_p$ represents the availability of the parking lot ‘$p$.’

Finally, whether there would be space available at the lot is another conflicting objective which is considered while selecting the route to a lot. The higher the vacancy in the parking lot, more it is likely to be selected, hence leading to minimization of the overall fitness function cost.

Constraints

$$d_{i,j} \geq 0 \forall i, j \in N; s_{i,j} \geq 0 \forall i, j \in N; x_p \geq 0 \forall p \in N$$

Equalities $d_{ij} = 0$ and $s_{ij} = 0$ hold for $i = j$. $x_p = 0$ when no parking is available.

The fitness function is modeled as the weighted sum of the above defined objective functions as follows:

$$\text{Min } f = w_1 f_1 + w_2 f_2 + w_3 f_3$$

(8)

where $\sum_{i=1}^{3} w_i = 1$.

A pilot survey was conducted among 50 drivers. The result of the survey was as follows (Table 1).

The weights generated by the Bayesian hierarchical method in (3) have been found to be much stable with respect to the error variance as compared to the classical approach. In fact, the improvement in error variance under small sample sizes is almost 100% by this method.

### 3.2 Genetic algorithm

The paper aims to provide a solution using genetic algorithm to solve multi-objective optimization problem. There are three factors which are considered: the distance, the speed and availability of the parking lot. The key components of GAs are: initial population of chromosomes, selection based on the fitness value, crossover to produce better descendants and mutation to produce randomness. The fitness function allocates a fitness value to every chromosome in the existing population. The fitter and better chromosomes are selected based on higher fitness values which are then selected for the crossover. Crossover involves combining the bits of one chromosome with the other chromosome to produce a child chromosome that inherits characteristics of both the father and the mother. Mutation is then performed to retain genetic diversity across generations. One such iteration is termed as a generation. Based on the principle of survival of the fittest, the algorithm controls which chromosomes should survive, die or reproduce.

The proposed algorithm is as follows:

Step 1: Generate randomly an initial population of N chromosomes.

Step 2: Compute the fitness value of all chromosomes using the fitness function.

Step 3: Perform tournament selection by selecting individuals from the population and insert into the mating pool.

Step 4: Perform single-point crossover with a crossover rate of 0.2

Step 5: Perform creep mutation where a random gene is selected and its value is changed with a random value between lower and upper bounds.

Step 6: Update population.

Step 7: If termination condition is met, then stop; else go to Step 2.

#### 3.2.1 Initial population

The first step involves producing an initial population. Each chromosome consists of a sequence of nodes starting from the start node, through one or multiple intermediate nodes terminating in the parking lot node. Every gene of the chromosome comprises a node identifier ensuring that the node does not reappear. A typical chromosome is represented as follows:

$$[0, 5, 22, 15, 25, 26, 2, 19, 30]$$

where $\{0\}$ is the starting node and marked red in Fig. 2, $\{30\}$ is the parking lot marked green in Fig. 2, and $\{5\}, \{22\}, \{15\}, \{25\}, \{26\}, \{2\}$ and $\{19\}$ are the intermediate nodes marked blue in Fig. 2.

#### 3.2.2 Selection (tournament selection)

An efficient and robust selection mechanism generally used by genetic algorithms is tournament selection. The procedure involves selecting fitter chromosomes and inserting into a mating pool. Chromosomes from this mating pool are then made to reproduce and generate new offspring for the
next generation. It is necessary that the mating pool contains ‘high-quality’ chromosomes. The technique conducts a tournament among S competitors, where S is called the tournament size. The chromosome with the best fitness value within the S competitors is selected as the winner and shifted to the mating pool which stores all the tournament winners. Considering a tournament size $S = 3$, three chromosomes are randomly selected from the population. Fitter among them is chosen as one of the parents. For example,

Chromosome 1: [0, 6, 22, 15, 25, 20, 18, 28]  
fitness = 0.728542

Chromosome 2: [0, 8, 7, 23, 3, 6, 22, 13, 25, 11, 2, 29]  
fitness = 0.859741

Chromosome 3: [0, 4, 22, 14, 13, 25, 11, 26, 2, 19, 17, 20, 18, 28]  
fitness = 0.958308

Chromosome 1, being the fittest among the three, gets selected as Parent 1. Since the problem deals with minimization, the lower the fitness value, fitter is the chromosome.

Chromosome 4: [0, 4, 22, 14, 13, 25, 11, 26, 2, 19, 17, 20, 18, 28]  
fitness = 0.958308

Chromosome 5: [0, 7, 4, 6, 22, 13, 1, 11, 2, 17, 21, 28]  
fitness = 0.960787

Chromosome 6: [0, 3, 6, 22, 15, 1, 11, 26, 21, 28]  
fitness = 0.827978

Here, Chromosome 6 is fittest and is selected as Parent 2. Thus, Chromosome 1 and Chromosome 6 are selected as Parent 1 and Parent 2 for crossover. The complexity of tournament selection is $O(n)$.

### 3.2.3 Crossover (single-point crossover)

The next step involves performing the crossover operation, and the performance of GA significantly depends on this crossover operators. New generation of chromosomes are created from the existing generation by crossing over the genetic information of the parents to produce a fitter child. Various crossover techniques exist that obtain optimum results over the generations. Single-point crossover has been used here where the two mating chromosomes are bisected at a single point and the halves are interchanged. An integer $m$ is assumed for each parent chromosome such that $m < n$, where $n$ is the length of the chromosome. The chromosomes are then bisected at the $m$th point, and genetic information is interchanged. Considering $m$ to be the midpoint of the length, each parent contributes 50% of its characteristics to the child. Suppose there are the following two parents:

Parent 1 (Chromosome 1): [0, 6, 22, 15, 25, 20, 18, 28]  
Parent 2 (Chromosome 6): [0, 3, 6, 22, 15, 1, 11, 26, 21, 28]

The length of parent $P_1$ is 8, so it contributes the first 50% of itself (first 4 genes) to the crossover process. The length of parent $P_2$ is 10, so it contributes the second half of itself (last 5 genes) to the crossover process. In this way the two children are created.

Child 1: [0, 6, 22, 15, 1, 11, 26, 21, 28]  
Child 2: [0, 3, 6, 22, 15, 25, 20, 18, 28]

The fitter of the two children is then shortlisted for the mutation process.

### 3.2.4 Mutation (creep mutation)

Mutation is a genetic operator implemented to provide genetic variety from one generation to the next by simply modifying one or multiple genes. Similar to crossover, there are various mutation techniques. Creep mutation has been used here where an arbitrary gene is chosen, and its information is replaced with random information within the bounds. The rate of mutation has been set as $1/n$ where $n$ is the length of the chromosome, i.e., only one gene in the entire chromosome is mutated. The gene to be mutated can be only replaced by a gene which will produce a valid path. Considering that
child 2 had been selected since it had a better fitness value than child 1, the input to the mutation process is as follows:

Child 2 (before mutation): [0, 3, 6, 22, 15, 25, 20, 18, 28]

A gene is selected at random, say {3}. This gene can be replaced by {4} since a valid path can be generated from {0} to {6} through {4}. Hence, the new chromosome is as follows:

Child 2 (after mutation): [0, 4, 6, 22, 15, 25, 20, 18, 28]

Transforming the population with crossover and mutation operator will usually take $O(NL)$ where $N$ is the size of the population and $L$ is the length of the chromosome.

### 3.3 Normalization of objectives

One has to normalize the objectives while formulating the composite objective function using the weighted sum method. As distance is in the range of kilometers, speed is in the range of kilometers/hours, and availability is a percentage value, there is this need to normalize them in the range $[0, 1]$ to bring them down to a common range for optimization. It is best to normalize them to a range $[0, 1]$. In order to scale a range $[a, b]$ to $[0, 1]$, the following function is used:

$$f(x) = \frac{x - a}{b - a}$$

### 3.4 Advantages and disadvantages of the proposed model

Bayesian hierarchical concept has been used to propose a route discovery technique for promoting sustainability. Through the implementation of this model in smart cities, the environmental conditions will gradually improve. The main focus of this work is on the improvement of the weights representing the relative importance of the possibly conflicting objective functions. Bayesian determination of weights finds high applicability in cases where conducting a large-scale survey is time-consuming, difficult to implement as well as expensive. This model works well when the sample size is relatively small. As the model is based on strong scientific theory, statistical analyses have reflected the advantages of the proposed model.

The proposed model uses the Bayesian framework which incorporates prior knowledge while determining the relative significance of the conflicting objectives it is expected to generate better results. This prior knowledge (distribution) is scientifically and mathematically improved starting with the pilot survey data and then applying Bayesian paradigm.

An advantage of using the sample data from the pilot survey makes the probability distribution narrower around the parameter’s true but unknown values. Another advantage of this technique is that this method can be applied to incorporate any number of objectives unlike the available continuation techniques.

The main disadvantage of this proposed model is that it focuses on the popular weighted sum technique which has been criticized by many for its drawbacks. The authors of this work claim that the criticisms have helped them propose this method which tries to overcome one such drawback of scientifically assigning weights to these objectives. Though not a disadvantage, the proposed system works effectively for smaller pilot surveys. For larger surveys, this model works as efficiently as the frequentist ones.

### 4 Results and discussion

Initially, the classical weighted sum technique has been implemented on 30 locations, numbered from 1 to 30 for reference. The distance matrix and the speed matrix are $[30] \times [30]$ matrices generated at run time by using the Google Maps API. The following three APIs have been used (a) Google Maps Distance Matrix API, (b) Google Maps Geocoding API and (c) Google Maps Places API. Thirty generations have been considered. The fitness values for the different epochs have been tabulated. It is observed that as the generations increases, the value of the fitness function tends to decrease till it stabilizes at an optimal value.

A plot has been made of fitness vs generation of the above values.

Thus, from Fig. 3 and its associated route table (Table 2) it is seen that the fitness value always exhibits a consistently decreasing trend as the number of generations increases across all time zones. Secondly, the routes as well as the parking lot vary depending on the time zone. This simulates a real-life scenario where parking lots and routes are bound to change as the values for the different factors changes as observed in Table 3. Although distance remains constant, the average speed and availability of parking lots change with time which gets finally reflected in the fitness function.

A comparative study has been performed for obtaining fitness values using the proposed Bayesian weights with respect to the existing frequentist approach for calculating the fitness values. Two different executions are displayed in Table 4 and plotted in Fig. 4. Both the executions confirmed that the Bayesian weights produced lower fitness values consistently.

On further analyzing the data to understand the effect of each of the parameters—distance, speed and parking availability, the following observations were noted.
### Table 2: Hour zone-wise fitness data across generations

| Gen | 12–4 a.m. | 4–8 a.m. | 8–12 p.m. | 12–4 p.m. | 4–8 p.m. | 8–12 a.m. |
|-----|-----------|----------|-----------|-----------|---------|-----------|
| 1   | 0.729     | 0.664    | 0.598     | 0.789     | 0.633   | 0.564     |
| 2   | 0.719     | 0.664    | 0.598     | 0.789     | 0.633   | 0.564     |
| 3   | 0.696     | 0.656    | 0.592     | 0.75      | 0.633   | 0.531     |
| 4   | 0.696     | 0.656    | 0.592     | 0.75      | 0.619   | 0.531     |
| 5   | 0.675     | 0.632    | 0.586     | 0.75      | 0.598   | 0.517     |
| 6   | 0.672     | 0.632    | 0.568     | 0.732     | 0.566   | 0.499     |
| 7   | 0.672     | 0.629    | 0.554     | 0.726     | 0.566   | 0.499     |
| 8   | 0.63      | 0.625    | 0.539     | 0.709     | 0.566   | 0.499     |
| 9   | 0.63      | 0.608    | 0.523     | 0.709     | 0.566   | 0.476     |
| 10  | 0.61      | 0.608    | 0.523     | 0.697     | 0.566   | 0.476     |
| 11  | 0.599     | 0.587    | 0.509     | 0.691     | 0.541   | 0.476     |
| 12  | 0.599     | 0.587    | 0.495     | 0.678     | 0.538   | 0.476     |
| 13  | 0.597     | 0.587    | 0.489     | 0.678     | 0.538   | 0.476     |
| 14  | 0.597     | 0.587    | 0.479     | 0.678     | 0.538   | 0.476     |
| 15  | 0.596     | 0.572    | 0.465     | 0.678     | 0.538   | 0.476     |
| 16  | 0.577     | 0.568    | 0.465     | 0.678     | 0.538   | 0.476     |
| 17  | 0.566     | 0.568    | 0.465     | 0.654     | 0.538   | 0.458     |
| 18  | 0.561     | 0.553    | 0.45      | 0.654     | 0.538   | 0.458     |
| 19  | 0.554     | 0.549    | 0.44      | 0.643     | 0.538   | 0.458     |
| 20  | 0.55      | 0.522    | 0.44      | 0.632     | 0.512   | 0.427     |
| 21  | 0.548     | 0.509    | 0.438     | 0.632     | 0.512   | 0.427     |
| 22  | 0.533     | 0.498    | 0.438     | 0.632     | 0.512   | 0.413     |
| 23  | 0.533     | 0.498    | 0.434     | 0.615     | 0.512   | 0.413     |
| 24  | 0.518     | 0.476    | 0.434     | 0.608     | 0.512   | 0.413     |
| 25  | 0.498     | 0.476    | 0.428     | 0.608     | 0.512   | 0.413     |
| 26  | 0.498     | 0.472    | 0.425     | 0.596     | 0.512   | 0.404     |
| 27  | 0.485     | 0.472    | 0.425     | 0.596     | 0.512   | 0.404     |
| 28  | 0.485     | 0.472    | 0.425     | 0.596     | 0.512   | 0.404     |
| 29  | 0.485     | 0.472    | 0.425     | 0.596     | 0.512   | 0.404     |
| 30  | 0.485     | 0.472    | 0.425     | 0.596     | 0.512   | 0.404     |

![Fig. 3 Plot of fitness versus generations for different time slots](image)

12 a.m.–4 a.m.: During these hours, the roads remain empty and the parking lots remain available. The average speed on all major roads is almost the same. So in this epoch distance has a higher priority over speed and parking lot availability. Routes with shorter distances were selected in this time zone.

4 a.m.–8 a.m.: This epoch marks a transition period. The later hours of this epoch mark the start of the peak hours when students go to school and corporates start moving toward their offices. Parking lots start filling up. Distance still has the highest priority here followed by speed and parking availability, and it is not dominated by a single parameter.

8 a.m.–12 p.m.: This marks the peak hours of the day. The roads remain congested. Parking lots start filling up fast. Thus, now speed has the highest priority followed by parking availability, followed by distance. Fastest routes featured in the result set more often showing the dominance of the speed parameter.

12 p.m.–4 p.m.: During this time, the parking lots are filled up. Congestion on roads reduces to some extent. As a result,
Table 3 Route table

| Time slot        | Routes                          |
|------------------|---------------------------------|
| 12 a.m.–4 a.m.   | [0, 4, 22, 14, 13, 11, 26, 21, 28] |
| 4 a.m.–8 a.m.    | [0, 4, 22, 13, 11, 2, 29]        |
| 8 a.m.–12 p.m.   | [0, 3, 4, 22, 13, 25, 20, 28]    |
| 12 p.m.–4 p.m.   | [0, 5, 22, 13, 11, 2, 29]        |
| 4 p.m.–8 p.m.    | [0, 7, 4, 22, 13, 25, 20, 28]    |
| 8 p.m.–12 a.m.   | [0, 5, 22, 13, 11, 2, 17, 21, 28] |

Table 4 Fitness values for Frequentist and Bayesian weights for two different runs

| Gen | Execution 1 | Execution 2 |
|-----|-------------|-------------|
|     | FREQ | BAYES | FREQ | BAYES |
| 1   | 0.612 | 0.599 | 0.718 | 0.679 |
| 2   | 0.6  | 0.577 | 0.718 | 0.679 |
| 3   | 0.581 | 0.544 | 0.718 | 0.64  |
| 4   | 0.558 | 0.532 | 0.671 | 0.64  |
| 5   | 0.558 | 0.522 | 0.657 | 0.64  |
| 6   | 0.541 | 0.522 | 0.628 | 0.622 |
| 7   | 0.501 | 0.487 | 0.628 | 0.596 |
| 8   | 0.501 | 0.472 | 0.628 | 0.579 |
| 9   | 0.484 | 0.455 | 0.612 | 0.579 |
| 10  | 0.484 | 0.455 | 0.612 | 0.567 |
| 11  | 0.458 | 0.429 | 0.612 | 0.561 |
| 12  | 0.458 | 0.429 | 0.605 | 0.548 |
| 13  | 0.434 | 0.427 | 0.605 | 0.548 |
| 14  | 0.434 | 0.411 | 0.586 | 0.518 |
| 15  | 0.418 | 0.411 | 0.586 | 0.518 |
| 16  | 0.418 | 0.388 | 0.586 | 0.518 |
| 17  | 0.418 | 0.388 | 0.586 | 0.494 |
| 18  | 0.402 | 0.351 | 0.573 | 0.494 |
| 19  | 0.402 | 0.344 | 0.573 | 0.483 |
| 20  | 0.402 | 0.34  | 0.573 | 0.472 |
| 21  | 0.388 | 0.338 | 0.573 | 0.452 |
| 22  | 0.388 | 0.323 | 0.519 | 0.452 |
| 23  | 0.388 | 0.323 | 0.519 | 0.435 |
| 24  | 0.366 | 0.308 | 0.507 | 0.428 |
| 25  | 0.366 | 0.288 | 0.493 | 0.428 |
| 26  | 0.359 | 0.288 | 0.493 | 0.416 |
| 27  | 0.341 | 0.275 | 0.493 | 0.416 |
| 28  | 0.333 | 0.275 | 0.493 | 0.416 |
| 29  | 0.333 | 0.275 | 0.493 | 0.416 |
| 30  | 0.333 | 0.275 | 0.493 | 0.416 |

Fig. 4 Plot of fitness versus generations for two different runs at time slot 4 a.m.–8 a.m. showing how fitness values change across 30 generations using frequentist (blue) and Bayesian (red) weights.

parking lot availability has the highest priority here followed by speed and then distance. Parking availability was the key parameter for route selection in this time zone.

4 p.m.–8 p.m.: This marks the time when people start returning to their houses. Parking lots start becoming empty. Here speed and distance more or less have the same priority. Parking lot availability has the least priority here. There was no marked observation of a single parameter dominating others.

8 p.m.–12 a.m.: This marks the night time. Roads start becoming empty, parking lots are almost empty, so again distance gains the maximum priority followed by speed and then parking availability. Distance again became the dominating parameter in this time zone.

5 Conclusion

A route discovery technique has been implemented based on Bayesian hierarchical theory. The work aims to promote sustainable development in a developing nation. Development of smart cities aids nations to improve their environmental conditions and help in maintaining a greener and cleaner world. This work relied on sound statistical techniques to improve the route optimization process for sustainable development through conservation of resources. The focus was on the improvement of the weights representing the relative importance of the possibly conflicting objective functions under consideration by proposing a flexible hierarchical Bayesian model. This technique has been analyzed for error variances, thereby quantifying the reliability of the estimates. Bayesian determination of weights finds high applicability in cases where conducting a large-scale survey is time-consuming, difficult to implement as well as expensive. When applied in the domain of route optimization in discovering the most suitable parking lot, the proposed methodology has produced results which display close resemblance to the phenomenon
observed in real-life situations. If implemented in reality, this would certainly ensure saving of time, energy and fuel, thus a greener world.

The paper presented a flexible hierarchical Bayesian technique that adequately accounts for various levels of uncertainties in estimating the weights of the conflicting objectives in a multi-objective optimization problem, whereas the objective of the existing approaches was to provide computationally feasible algorithms, focusing on the solution set instead of the weights. Since the goals are different, different performance criteria measures the performances of the approaches. Thus, the proposed methodology is not directly comparable with the existing methodologies. This work has the advantage of using the Bayesian framework which incorporates prior knowledge (echoed through the prior distribution) while determining the relative significance of the conflicting objectives. This prior knowledge (distribution) is scientifically and mathematically improved using the pilot survey data following Bayesian paradigm. An advantage of using the sample data from the pilot survey makes the probability distribution narrower around the parameter’s true but unknown values. Unlike the available continuation techniques, this method can be applied with convenience to handle any number of objectives. The weighted sum method has been used here to solve the problem using genetic algorithm.

5.1 Future scope of improvement

The proposed model may be expanded further by considering Pareto optimal solutions by plotting the Pareto curve and obtaining solutions from the Pareto front. As this method yields a posterior probability distribution over the weights, the stochastically generated weight vectors can be used to obtain the points on the Pareto front with less computational complications. The system may be improved by varying the various GA parameters such as population size, number of generations and mutation rate. A prototype model designed above has helped us to study the impact of each of the conflicting objectives’ dominance at different hours of the day and how their priorities change over time. The system has been implemented with thirty-one locations which can be increased to a bigger domain. The system may be further improved to work with dynamic data reflecting the changing objective parameters during different hours of the day. However, the limitation remains that the proposed methodology will work efficiently for small sample sizes. As the sample size grows, the efficiency of the system gradually decreases and tends to generate similar weights as produced by the frequentist technique.

References

1. Ombuki B, Ross BJ, Hanshar F (2006) Multi-objective genetic algorithms for vehicle routing problem with time windows. Appl Intell 24:17–30
2. Coello Coello CA, Van Veldhuizen DA, Lamont GB (2002) Evolutionary algorithms for solving multi-objective problems. Springer US, New York
3. Fonseca CM, Fleming PJ (1995) An overview of evolutionary algorithms in multi-objective optimization. Evol Comput 3:1–16
4. Van Veldhuizen DA, Lamont GB (2000) Multi-objective evolutionary algorithms: analyzing the state-of-the-art. Evol Comput 8:125–147
5. Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, New York
6. Goldberg DE (1998) Genetic algorithm in search, optimization, and machine learning. Addison-Wesley, Reading
7. Haupt RL, Haupt HS (2004) Practical genetic algorithms, 2nd edn. Wiley, Hoboken
8. Arulmozhiyal R, Jabril AM (2012) A nonlinear weights selection in weighted sum for convex multi-objective optimization. Facta Univ Ser Math Inform 27:357–372
9. Zadeh LA (1963) Optimality & non-scalar-valued performance criteria. IEEE Trans Autom Control 8:59–60
10. Konak A, Coit DW, Smith AE (2006) Multi-objective optimization using genetic algorithms: a tutorial. Reliabil Eng Syst Saf 91:992–1007
11. Athan TW, Papalambros PY (1996) A note on weighted criteria methods for compromise solutions in multi-objective optimization. Eng Opt 27:155–176
12. Ryu JH, Kim S, Wan H (2009) Pareto front approximation with adaptive weighted sum method in multi-objective simulation optimization. In: Proceedings of 2009 Winter Simulation Conference, Austin, TX, pp 623–633
13. Schmaranzer D, Braune R, Doerner KF (2019) Multi-objective simulation optimization for complex urban mass rapid transit systems. Ann Oper Res 1–38
14. Dantzig GB, Ramser JH (1959) The truck dispatching problem. Manag Sci 6(1):80–91
15. Carać T, Galić A, Fosin J, Gold H, Reinholz A (2008) A modelling and optimization framework for real-world vehicle routing problems, vehicle routing problem. I-Tech Education and Publishing, Vienna
16. Golden B, Raghavan S, Wasil E (2008) The vehicle routing problem: latest advances and new challenges. Operations research-computer science interfaces series. Springer, Berlin, p 43
17. Han Y, Shan J, Wang M, Yang G (2017) Optimization design and evaluation of parking route based on automatic assignment mechanism of parking lot. Adv Mech Eng 9:1–9
18. Siemens (2020) Intelligent Parking Solutions—Siemens. www.mobility.siemens.com/global/en/portfolio/road/parking-solutions/intelligent-parking-solutions.html. Accessed 23 June 2020
19. Xiao Y, Konak A (2017) A genetic algorithm with exact dynamic programming for green vehicle routing & scheduling problem. J Clean Prod 167:1450–1463
20. Lee HY, Shin H, Chae J (2018) Path planning for mobile agents using a genetic algorithm with a direction guided factor. Electronics 7:212
21. Fogel D (2005) Evolutionary computation: toward a new philosophy of machine intelligence, 3rd edn. Wiley, Hoboken
22. Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence. MIT Press, Cambridge

Funding  This work is not funded by any agency/organization.
23. Bryant K (2000) Genetic algorithms and the traveling salesman problem. Master’s Dissertation. Harvey Mudd College, Claremont, United States
24. Chand P, Mohanty JR (2013) A Multi-objective vehicle, routing problem using dominant rank method. In: Proceedings of international conference in Distributed Computing & Internet Technology, International Journal of Computer Applications 0975-8887, pp 29–34
25. Pelikan M (2010) Genetic algorithms. MEDAL Report No. 2010007
26. Athan TW, Papalambros PY (1996) A quasi-Monte Carlo method for multi-criteria optimization. Eng Opt 27:177–198
27. Gennert MA, Yuille AL (1998) Determining the optimal weights in multiple objective function optimization. In: 2nd International Conference on Computer Vision, IEEE, Los Alamos, CA, pp 87–89
28. Martins MSR, Delgado MRBS, Lüders R (2018) Hybrid multi-objective Bayesian estimation of distribution algorithm: a comparative analysis for the multi-objective knapsack problem. J Heuristics 24:25–47
29. Tang Y, Marshall L, Sharma A, Ajami H (2018) A Bayesian alternative for multi-objective eco-hydrological model specification. J Hydrol 556:25–38
30. Wada T, Hino H (2019) Bayesian optimization for multi-objective optimization and multi-point search. ArXiv, abs/1905.02370
31. Brochu E, Cora VM, de Freitas N (2010) A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv:1012.2599
32. Hernández-Lobato D, Hernandez-Lobato J, Shah A, Adams R (2016) Predictive entropy search for multi-objective bayesian optimization. In: International conference on machine learning, pp 1492–1501
33. Tsoulos IG, Stavrou V, Mastorakis NE, Tsalikakis D (2019) GenConstraint: a programming tool for constraint optimization problems. SoftwareX 10:100355
34. Mockus J (1989) Bayesian approach to global optimization: theory and applications. Springer, Dordrecht
35. Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems, vol 25, pp 2951–2959
36. Tajbakhsh SD (2012) A fully Bayesian approach to the efficient global optimization algorithm. Ph.D. Dissertation. The Pennsylvania State University, Pennsylvania
37. Galuzio PP, Hochsteiner de Vasconcelos Segundo E, Santos-Coelho LD, Mariani VC (2020) MOBOpt—multi-objective Bayesian optimization. SoftwareX 12:100520

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.