

Data-Driven Surrogate-Assisted Multi-Objective Evolutionary Optimization of A Trauma System

Handing Wang, *Member, IEEE*, Yaochu Jin, *Fellow, IEEE*, Jan O. Jansen

Abstract—Most existing work on evolutionary optimization assumes that there are analytic functions for evaluating the objectives and constraints. In the real-world, however, the objective or constraint values of many optimization problems can be evaluated solely based on data and solving such optimization problems is often known as data-driven optimization. In this paper, we divide data-driven optimization problems into two categories, i.e., off-line and on-line data-driven optimization, and discuss the main challenges involved therein. An evolutionary algorithm is then presented to optimize the design of a trauma system, which is a typical off-line data-driven multi-objective optimization problem, where the objectives and constraints can be evaluated using emergency records only. As each single function evaluation involves large amount of patient data, we develop a multi-fidelity surrogate management strategy to reduce the computation time of the evolutionary optimization. The main idea is to adaptively tune the approximation fidelity by clustering the original data into different numbers of clusters and a regression model is constructed to estimate the required minimum fidelity. Experimental results show that the proposed algorithm is able to save up to 90% of computation time without much sacrifice of the solution quality.

Index Terms—data-driven optimization, multi-objective optimization, evolutionary algorithm, surrogate, clustering, trauma system design.

I. INTRODUCTION

Evolutionary algorithms (EAs) are a class of nature-inspired global optimization algorithms. They distinguish themselves with traditional mathematical programming algorithms in that they do not require analytical models of the objective and constraint functions of the problem to be optimized and are less vulnerable to local optimums. For these reasons, EAs have enjoyed great success in solving a wide range of industrial problems [1], [2].

Although they do not require analytical objective or constraint functions, most EAs assume that analytical mathematical functions are available for evaluating the objectives and assess the constraints. Unfortunately, many real-world optimization problems do not have analytic objective or constraints functions and optimization can be pursued only based on data

collected from physical experiments, real events, or complex numerical simulations. Solving optimization problems solely based on data is known as data-driven optimization.

Most existing work on data-driven evolutionary optimization has actually been carried out in the area of surrogate-assisted evolutionary optimization, where fitness evaluations reply on very expensive physical experiments or highly time-consuming computer simulations, such as in drug design [3] and aerodynamic shape optimization [4], [5]. In addition, evolutionary optimization involving subjective human evaluations, which is known as interactive evolutionary optimization [6], [7], also falls in this type of data-driven optimization. We term this type of data-driven optimization *on-line data-driven optimization*. One main feature of on-line data-driven optimization is that during the optimization, it is still possible to acquire new data, although this might be costly or computationally expensive. For on-line data-driven optimisation, most existing surrogate-assisted EAs [8] are applicable.

By contrast, another type of data-driven optimization problems have largely been overlooked so far. In this type of optimization problems, no data can be collected per conducting additional experiments or computer simulations during the optimization, as each data sample may be an event that accidentally occurs. For example, design of trauma systems can theoretically be carried out based on comprehensive geospatial analysis. In practice, however, optimization of trauma system design can be accomplished using emergency records in the previous years as an approximate geospatial guidance, as reported in [9], [10], where a multi-objective evolutionary algorithm (MOEA) has been employed to optimize the trauma system of Scotland. In such problems, the objective values may vary with the number of emergency records and a larger number of the emergency records can lead to richer geospatial information, and consequently more accurate optimization. Note however, that new records cannot be actively collected during the optimization process. We term such data-driven optimization *off-line data-driven optimization*.

Data-driven optimization usually needs to address multiple conflicting objectives, like in most real-world optimization problems. Such problems are known as multi-objective optimization problems (MOPs) [11]. Over the past two decades, a large number of multi-objective evolutionary algorithms (MOEAs) have been developed [12], which can be largely classified into three groups, i.e., dominance based, aggregation based, and performance indicator based. Dominance based MOEAs [13], [14] rely mainly on dominance comparisons for selecting parents [15]; aggregation-based MOEAs, also know as decomposition based methods, optimize a set of single-

This work was supported in part by an EPSRC grant (No. EP/M017869/1) on "Data-driven surrogate-assisted evolutionary fluid dynamic optimisation", in part by the National Natural Science Foundation of China (No. 61271301) and in part by the Joint Research Fund for Overseas Chinese, Hong Kong and Macao Scholars of the National Natural Science Foundation of China (No. 61428302).

H. Wang is with the Department of Computer Science, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: wanghanding.patch@gmail.com).

Y. Jin is with the Department of Computer Science, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: yaochu.jin@surrey.ac.uk).

J. O. Jansen is with NHS Grampian, Health Services Research Unit, University of Aberdeen, Aberdeen, AB25 2ZD, U.K. (e-mail: jan.jansen@abdn.ac.uk).

objective sub-problems with a pre-assigned weight set [16], [17]; and indicator-based MOEAs employ a performance indicator as their selection criterion [18]. However, little research work dedicated to addressing challenges in data-driven multi-objective evolutionary optimization has been reported so far, in particular off-line data-driven evolutionary optimization in the presence of huge amount of data.

This paper first provides a categorization of data-driven optimization problems and discusses challenges and opportunities in solving these problems using evolutionary algorithms. To the best of our knowledge, an elaborated categorization and discussion of data-driven evolutionary optimization are still missing in the literature. Then, the paper focuses on evolutionary design of a trauma system, a typical off-line data-driven bi-objective optimization problem in the presence of large amount of data [9]. Such optimization problems are computationally very intensive, since the objective as well as the constraint functions are evaluated based on a large number of emergency records in the previous. As it has been indicated that the emergency records are often distributed in special patterns [19], this work aims to reduce the amount of computation time by grouping the data into clusters and then using the cluster centers as representative data points for function evaluations. Consequently, the fewer clusters the data is grouped into, the faster the calculation will be, and the less accurate the evaluations are. Thus, the key challenge is how to adaptively balance the reduction of computation time and the accuracy of the estimated function values based on the clustered data.

The rest of this paper is organized as follows. We discuss in more detail the challenges in data-driven evolutionary optimization in Section II. The data-driven trauma system design optimization problem is introduced in Section III, followed by a description of an adaptive multi-fidelity surrogate management technique in Section IV. Empirical results are presented and discussed in Section V. Section VI concludes the paper.

II. DATA-DRIVEN EVOLUTIONARY OPTIMIZATION

A generic framework for data-driven evolutionary optimization is in Fig. 1. At each generation, EAs start with a population of candidate solutions (parents), and generate offspring solutions by applying variation operators, such as crossover, mutation, local search to the parent solutions. All offspring solutions are evaluated according to the objective constraint functions before the parent individuals for the next generation can be selected. For data-driven evolutionary optimization, evaluation of the objectives and constraints are based on data.

As can be seen in Fig. 1, the main difference between data-driven evolutionary optimization and evolutionary optimization using analytical objective and constraint functions lies in function evaluations. In some cases, experimental or simulation data need to be preprocessed before they can be applied to function evaluations. More often than not, acquisition of data is either costly or computationally intensive, seriously limiting the number of function evaluations. One widely adopted technique to achieve acceptable solutions using a small number

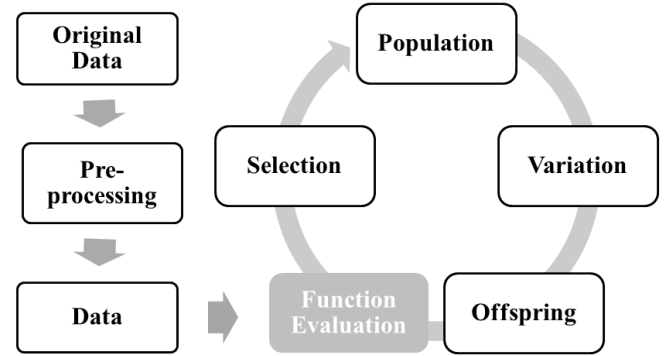


Fig. 1. A generic framework of data-driven evolutionary algorithms.

of function evaluations is to use surrogates, also known as metamodels or approximate function evaluations [8], [20] to replace in part the exact function evaluations in optimization. Management of the surrogate, including when to use and update the surrogates, plays a key role in surrogate-assisted optimization [21].

In the following, we elaborate the differences in managing surrogates in off-line and on-line data-driven evolutionary optimization.

A. Off-line and On-line Data-driven Optimization

Because only a very small number of exact function evaluations can be afforded, data-driven EAs usually resort to surrogates to reduce the needed number of expensive function evaluations. Depending on whether it is off-line or on-line data-driven optimization, the surrogate management strategies can vary, as illustrated in Fig. 2 and Fig. 3, respectively.

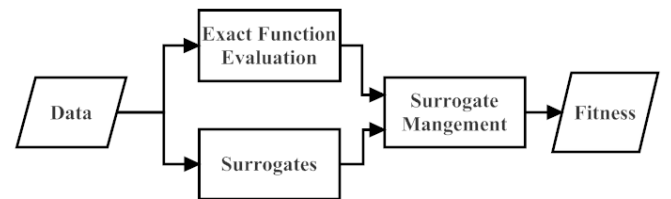


Fig. 2. Surrogate management in off-line data-driven evolutionary optimization.

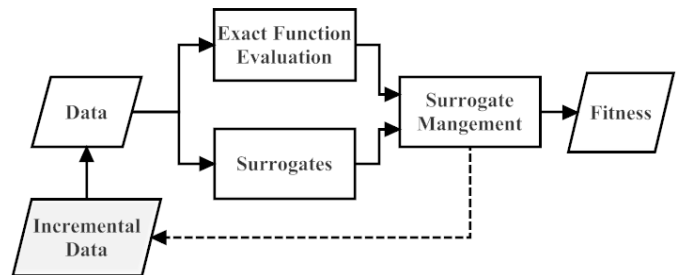


Fig. 3. Function evaluations of on-line data-driven EAs.

From Fig. 2, we can see that in off-line data-driven optimization, data is acquired before evolutionary optimization.

During the optimization, the EA can explore the give data as much as possible, but no new data will be made available during the optimization. By contrast, in on-line data-driven evolutionary optimization, new data can be generated during optimization, as shown in Fig. 3. However, the incremental data, those obtained during the on-line data-driven evolutionary optimization, may or may not be controlled by surrogate management. If the surrogate management strategy can actively sample new data at the desired points, the accuracy of the surrogates can be most effectively enhanced. However, in some specific cases, the surrogate management strategy does not have any control on the incremental data. This difference in controlling the incremental data generation is manifested by the dash line connecting the surrogate management block and the incremental data in 3.

As a whole, data-driven evolutionary optimization can be largely divided into three categories, namely, off-line data-driven optimization where no new data can be generated during optimization, on-line data-driven optimization with uncontrolled incremental data, and on-line data-driven optimization with controlled incremental data.

B. Surrogate Models and Surrogate Management

To reduce the needed number of expensive function evaluations, surrogate-assisted evolutionary algorithms (SAEAs)[21], [8], [20] can be used, where the main idea is to use one [22], [23] or multiple surrogates [24], [25], [26] to approximate the expensive function evaluation globally or locally [27], [28], [29]. Note that an implicit assumption here is that the computational cost for constructing and using the surrogates is much less than that for fitness evaluations using the original expensive function.

Many machine learning models, such as linear, nonlinear or polynomial repression models [30], Kriging or Gaussian processes [31], [32], [33], [34], [35], support vector machines (SVMs) [36], radial basis function (RBF) networks [37], [38], [39], and many other neural networks [40], [41], [42], [43], [44]. Several ideas have been proposed for choosing individuals to be re-evaluated using the original objective functions, which is one key issue in surrogate management. These include selecting potentially good solutions [45], [44], selecting representative solutions [24], [46], or selecting solutions with large amount of uncertainty [47], [48]. As an approximation of the original objective function, surrogate models are subject to errors. It has been indicated that errors introduced by surrogates are acceptable so long as they do not mislead the search [49], and in some cases they can even be exploited [49], [50]. Therefore, how to manage the trade-off between the accuracy and the number of exact function evaluations is the main issue in SAEAs. Metrics for measuring the quality of surrogates have been proposed in [49], [51].

It should be noted that most surrogate management techniques in the literature are proposed for on-line data driven evolutionary optimization where the model management strategy has control over the incremental data.

C. Challenges

In the following, we discuss the main challenges to data-driven evolutionary optimization from the perspectives of computational cost, data quantity, data quality, heterogeneity of the data, all of which are related to surrogate construction.

- **Computational Cost:** Exact function evaluations in most data-driven optimization problems are expensive due to various reasons. For instance for problems based on physical experiments or computational simulations, each function evaluation either reply on expensive physical experiments such as wind tunnel tests or crash test, or complex and highly time-consuming numerical simulations such as computational fluid dynamics simulations [5]. For most off-line data-driven optimization problems, such as design of trauma systems [9], [10], it often takes huge amount of time to collect a large number of emergency records, so do a precise evaluation of candidate designs. Evolutionary optimization of such problems will not be able to afford thousands of function evaluations as required by most existing EAs.
- **Data quantity:** Construction of surrogates, either computational models or any estimate of the exact function evaluations, is indispensable in data-driven optimization due to the high computational cost as mentioned above. However, construction of surrogates is often challenged by the amount, quality and distribution. Either a too small size or a very large size of training data samples will create difficulties for training surrogates. Lack of training data will it impossible to build accurate surrogates when the dimension of the decision space is high [5], while large amount of training may result in increasing computational cost [52], [53].
- **Data quality:** Quality of the data includes the distribution of the data and the amount of uncertainty in the data. It is very likely that the data is ill-distributed, imbalanced [54], incomplete [55], [56], and is contaminated by noise [57], [58]. In some cases, search for robust optimal solutions will be of great practical significance [59], [60].
- **Heterogeneity of data:** Finally, surrogate construction can be further complicated by the nature of the data. In other words, training data might come from multiple sources, presenting in a combination of heterogeneous forms such as numerical data, texts and images [61]. How to fuse the heterogeneous data may considerably affect the quality of the function evaluations and consequently the quality of the constructed surrogates [62].

To alleviate the difficulties resulting from the quality and heterogeneity of the data, properly pre-processing the data may be helpful. Most machine learning and data mining techniques can be adopted in the pre-processing data, such as data clustering [63], [64] for reducing the size of data, principal component analysis [] and feature selection [65] for dimension reduction, and and regression techniques [66] for learning the relationship in the data.

III. DATA-DRIVEN MULTI-OBJECTIVE OPTIMIZATION OF A TRAUMA SYSTEM

A trauma system is a network of hospitals, stratified by their clinical capability, and supported by emergency medical services [67]. The aim of a trauma system is to reduce death and disability from injury, by matching patients needs with hospitals resources. This concept is sometimes referred to as "right care, right time, right place" [68]. A well-designed trauma system should be both efficient and economical, i.e., providing the most effective treatment with the lowest resource cost [69]. On one hand, every patient should be treated with suitable health care. On the other hand, the resources including trauma centers, surgeons, ambulances, and medicines should be saved. Therefore, facilities and locations of trauma centers and the geospatial injury density affect the behavior of the whole trauma system [9]. Thus, the trauma system design problem becomes an optimization problem based on the geospatial information to search for the optimal configuration of trauma centers.

However, the geospatial information is hard to be measured directly. The emergency records within a long time period can play that role indirectly, because the distribution of emergency records implicitly shows the geospatial injury density. The emergency records can be used to simulate the trauma system that is presented by one configuration, which can be evaluated by the simulated output (both clinical and final) of those emergency records. Thus, the function evaluation of the trauma system design problem relies on data, which makes it a data-driven optimization problem.

A. Allocation Algorithm for Patients

The trauma system of Scotland has three kinds of trauma centers with different levels of capability [70], including major trauma centre (MTC), trauma unit (TU), and local emergency hospital (LEH). An MTC provides clinical services for patients with major trauma. A TU handles less severely injured patients. An LEH manages patients with minor injuries. In the data, patients are classified into two groups, i.e., patients triaged to MTC or to TU. A patient will be sent to a center with suitable capability according to the nature of the patient's injury [71]. The priority of the patient allocation is shown Fig. 4.

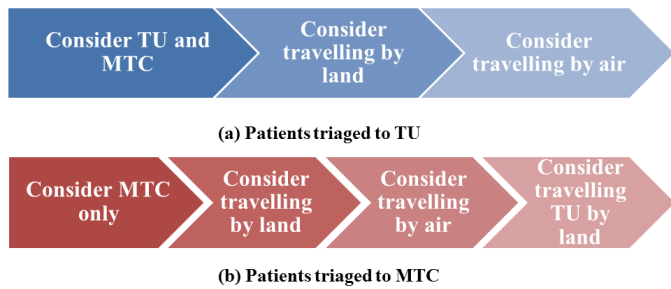


Fig. 4. Priority in the allocation algorithm for patients triaged to TU or MTC.

For a patient who is triaged to a TU, the allocation algorithm first considers travelling by land and chooses the center with

the shortest drive time (either a TU or an MTC). If the time of the shortest land transport is not acceptable, the allocation algorithm considers travelling by air and chooses the center with the shortest flight time (either a TU or an MTC). However, if the flight time is longer than the drive time, the allocation algorithm chooses to travel by land.

For a patient who is triaged to an MTC, the allocation algorithm considers the nearest MTC by land at first. If the drive time to the MTC is not acceptable, it considers travelling to the MTC by air. If the flight time to the MTC is still not acceptable, the allocation algorithm will consider a TU with shorter travel time by land, thus the level of facilities in the TU will be lowered for that patient.

In the data, the location and injury of every patient are recorded. Using the allocation algorithm, patients are triaged to a suited trauma center. Taking a particular patient (patient i) as an example, the patient is assigned to a matched trauma center according to the accident location and severity of injury. The allocation algorithm will accordingly output information such as the travel time, assigned center, and helicopter usage, as shown in Fig. 5, where T_i is the traveling time from the location of the i -th patient to the assigned center. There is an exceptional situation where the patient should have been triaged to an MTC but had to be sent to a TU because of an unaccepted long travel time to MTC. This is indicated by the i -th bit (L_i) of a binary string L (1 for an MTC exception and 0 for not). The binary string H_i is of length of n_d (the number of helicopter depots) and the j -th ($1 \leq j \leq n_d$) bit of H_i is set to 1 if the i -th patient is assigned to the j -th helicopter depot. In other words, all bits of H_i are set to 0 if patient i is sent by land. The binary string G_i is of length n_{MTC} (the number of MTCs in the configuration), and the j -th ($1 \leq j \leq n_{MTC}$) will be set to 1 if the i -th patient is sent to the j -th MTC, otherwise to 0. Thus, all bits of G_i are set as 0 if the patient is sent to a TU. Table I shows an example of the output of the allocation algorithm for patient i , from which we can see that this patient was sent to the 4-th MTC by air from the 2nd helicopter depot, which took 25 minutes.

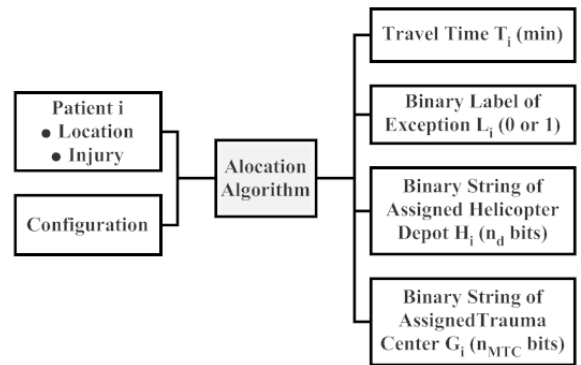


Fig. 5. Outputs of the allocation algorithm for one patient

B. Problem Formulation

The basic aim of a trauma system is to provide the good and quick treatment of all patients. However, it is impossible

TABLE I
EXAMPLE OUTPUTS OF THE ALLOCATION ALGORITHM FOR PATIENT i .

Output	Length	Value	Meaning
T_i	NA	25 min	Travel time equals to 25 min
L_i	1	0	Non-MTC exception
H_i	$n_d = 3$	010	The 2nd depot is used
G_i	$n_{MTC} = 4$	0001	Patient i is sent to the 4th MTC

to build a large number of trauma centers or send all patients by air given the limited resource. There are conflicts between the financial and clinical outcomes in the design of a trauma system. From the perspective of a patient, the travel time from the location of the accident to the trauma center is the most important. The time should be as short as possible, because the shorter time definitely reduces the chance of death or disability. Increasing the numbers of TU and MTC decreases the travel time of patients. However, the budget for a trauma system is limited. In addition, MTC should handle as many patients as possible to gain experiences for its further improvement. Finally, to reduce the transport cost, the helicopter usage should also be reduced. From the above discussions, we can see that many targets in designing a trauma system are conflicting with each other.

To date, 18 trauma centers in total have been built in Scotland. The design problem is to search for the optimal configurations of these centers to achieve both good financial and clinic outcomes. Different configurations result in different situations for these conflicting targets discussed above and it is usually hard to find a configuration that can simultaneously satisfy all these targets. Essentially, the aim of designing a geospatially optimized trauma system in Scotland is to find an optimized configuration of centers achieving the preferred trade-off between the conflicting targets [9]. In other words, the task is to assign different levels (MTC, TU, and LEH) to the existing centers, which can be seen as a combinatorial MOP [11] as described in the following subsections.

1) *Objectives*: As discussed above, design of a trauma system involves in many conflicting objectives. For simplicity, the problem is formulated as a bi-objective optimization problems, both related to the interest of patients. The first is to minimize the total travel time of all the patients, the other one is to minimize the number of exceptions, where a patient should have been sent to an MTC but was eventually sent to a TU because of the unaccepted long travel time needed to an MTC. The total travel time of all patients in the records can be described by Equation (1), where T_i is the travel time of patient i and N is the number of patient in the data:

$$f_1 = \sum_{i=1}^N T_i \quad (1)$$

The second objective, i.e., the number of exceptions can be calculated by Equation (2), where L_i records whether patient i is an exception, as shown in Fig. 5:

$$f_2 = \sum_{i=1}^N L_i. \quad (2)$$

2) *Constraints*: Although the financial aspects are not as important as patients, they impose the main limits for designing the trauma system. Therefore, helicopter usage, MTC case volume, and TU proximation are set as the constraints of the trauma system design problem.

The capacity of the j -th emergency depot is n_h^j helicopter transfers everyday. Therefore, the constraint of helicopter usage of the j -th emergency depot should satisfy Equation (3), where D is the number of days and H_i^j is the j -th bit of H_i recording the usage helicopter depot for the i -th patient.

$$h_j = \sum_{i=1}^N H_i^j \leq D n_h^j, 1 \leq j \leq n_d \quad (3)$$

In a trauma system, an MTC has more facilities than a TU and therefore is expected to deal with more cases to gain experiences [72]. The number of patients sent to the j -th MTC everyday in the configuration should be higher than a threshold V . Thus, the constraints of the case volume of the j -th MTC is set as Equation (4), where G_i^j is the j -th bit of G_i recorded for the i -th patient as shown in Fig. 5.

$$g_j = \sum_{i=1}^N G_i^j \leq D V, 1 \leq j \leq n_{MTC}. \quad (4)$$

To control the number of TUs in the trauma system, the TUs should not locate too close to each other. Therefore, the distance between any two TUs should not be smaller than a predefined minimum distance d_{TU} :

$$d_{jk} \geq d_{TU}, 1 \leq j \leq n_{TU}, 1 \leq k \leq n_{TU}, j \neq k, \quad (5)$$

where d_{jk} is the distance between the j -th and k -th TU, $1 \leq j, k \leq n_{TU}$.

In summary, for the configuration with n_{MTC} MTCs and n_{TU} TUs, there would be $n_d + n_{MTC} + C_{n_{TU}}^2$ constraints in total. Of them, $n_d + n_{MTC}$ constraints (h and g) are data-driven.

C. Multi-Objective Evolutionary Algorithm

According to the discussions in Section II, we can see that design of the trauma system is an off-line data-driven bi-objective optimization problem, as no incremental data can be made available during the optimization.

In the following, we present the representation of the configuration of a trauma system for evolutionary optimization.

1) *Genetic coding*: The 18 trauma centers in Scotland are encoded in the following order: Glasgow Royal, Aberdeen Royal Inf, Ninewells, D&G Royal Inf, Edinburgh Royal, Forth Valley, Hairmyres, Raigmore, Crosshouse, Wishaw, Royal Alexandra, Borders, Southern General, Ayr, Victoria Kirkcaldy, Monklands, Inverclyde, and Perth. We use an integer between 0 and 2 to encode whether a particular center should be set as an MTC, or a TU, or an LEH, encoded by "2", "1", or "0", respectively. Since there are 18 centers in the current trauma system, the length of the chromosome (representing one candidate configuration) is 18. For example, the chromosome [11112011100100101] means that Glasgow Royal is set to an MTC, Aberdeen Royal Inf, Ninewells, D&G

Royal Inf, Edinburgh Royal, Forth Valley, Hairmyres, Raigmore, Crosshouse, Wishaw, Royal Alexandra, and Borders are set to TUs, and Southern General, Ayr, Victoria Kirkcaldy, Monklands, Inverclyde, and Perth are set to LEHs.

Based on the above coding for evolutionary design of the trauma system, NSGA-II is adopted for optimizing the bi-objective optimization problem. For reproduction, 3-point crossover with a probability of 1 and point mutations with a probability of 0.2 are employed.

In addition to the genetic variations, variable neighborhood search (VNS) [73], a local search strategy particularly developed for solving combinatorial optimization problems, has also been employed.

The main idea of VNS is to search in a neighborhood starting from a local optimum of another neighborhood. The main reason is that for combinatorial optimization problems, it is easy to get trapped in a local optimum if the local search is performed only in one neighborhood. It has been shown that a large variety of neighborhoods helps improve diversity in local search [74]. Therefore in VNS [75], several different neighborhoods are used. Starting from a local optimum, VNS traverses through one neighborhood to obtain a new local optimum, from where it search within another neighborhood.

VNS is applied only to the non-dominated solutions at every generation. To adapt VNS to trauma system design, three neighborhoods NB_1 , NB_2 , and NB_3 have been defined. In the neighborhood NB_1 , a gene is randomly chosen, and then other genes are all swapped with that gene to generate new solutions. In the neighborhood NB_2 , a gene is randomly chosen, then the value of the randomly chosen gene is changed to all other feasible values to generate new candidate solutions. In the neighborhood NB_3 , two genes are randomly chosen, then all the combinational feasible values of those two bits are traversed to generate new configurations. **This is unclear.**

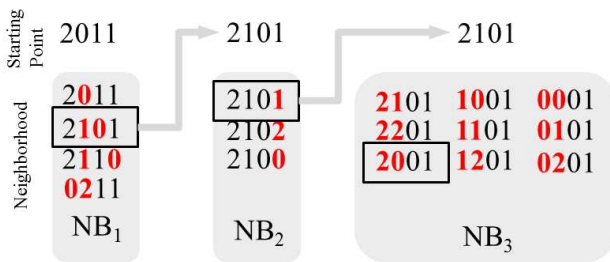


Fig. 6. An illustrative example of VNS for design of a trauma system with four centers.

Figure 6 provides an illustrative example of VNS for design of a trauma system with four centers, each candidate design consisting of four genes whose value can be "0", "1", or "2". Of the three neighborhoods, NB_1 and NB_2 contain three feasible solutions, and NB_3 nine solutions. Assume solution [2011] is a non-dominated solution in the current population and the 2-nd gene of this solution is randomly chosen. Then all the feasible solutions in NB_1 are searched starting from this one. Now assume [2101] dominates [2011], which makes it the starting point to perform local search in NB_2 . In NB_2 , assume the last gene is chosen at random, after performing the

local search in NB_2 , no solution is able to dominate [2101]. Consequently, [2101] remains the starting point for performing search in NB_3 . According to the local search strategy for NB_3 , two gene, e.g., the 1st and 2nd genes are chosen at random. If among those solutions in NB_3 , solution [2001] dominates all other solutions, it will be the final output of the VNS.

IV. DATA-DRIVEN MULTI-FIDELITY SURROGATES FOR MULTI-OBJECTIVE OPTIMIZATION

Data-driven optimization of the trauma system using NSGA-II has been reported in [10]. In that work, each objective and constraint evaluation involves all patients in the data, which makes the evolutionary optimization very time-consuming. In fact, one evolutionary run using NSGA-II based on the data collected within one year takes about one day and if more data are to be used, the computation time will become prohibitive. In this work, we propose a multi-fidelity surrogate management strategy dedicated to a class of off-line data-driven optimization problems involving huge amount of data to reduce the computation time without degrading the performance of the solutions.

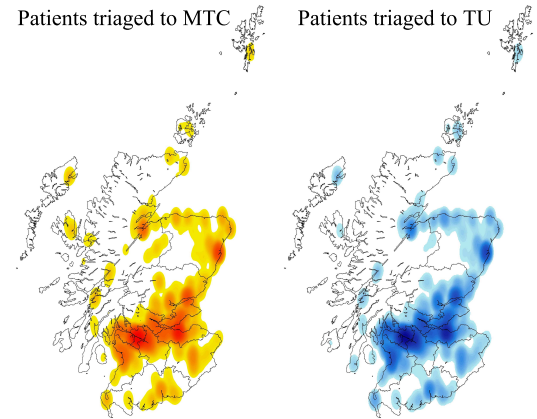


Fig. 7. Density maps of patients triaged to MTC and TU.

Optimization of the trauma system studied in this work relies on about 40,000 emergency records collected within one year. Each record includes information about whether the victim is triaged to MTC or TU, the location of the accident and the travel time to all the centers by land and air. It has been found that most records are distributed with particular patterns [19], which is related to the population density as shown in Fig. 7. Intuitively, if we cluster the records into K groups, where $1 \leq K \leq N$, where N is the number of records (patients) in total. Let C_i and N_i denote the center of the i -th cluster and the number of records in the cluster, respectively, where $1 \leq i \leq K$. Once the data is clustered, only one record closest to C_i , instead of all N_i records in the i -th cluster will be used for objective and constraint evaluations. Thus, the number of clusters (K) will determine the fidelity of the objective and constraint evaluations, and the smaller K is, the less accurate the evaluations will be. In this case, the two objectives and the first two constraints in Equations (1)-(5) can be approximated

calculated as follows.

$$\hat{f}_1 = \sum_{i=1}^K N_i T_i \quad (6)$$

$$\hat{f}_2 = \sum_{i=1}^K N_i L_i \quad (7)$$

$$\hat{h}_j = \sum_{i=1}^K N_i H_i^j, 1 \leq j \leq n_d \quad (8)$$

$$\hat{g}_j = \sum_{i=1}^K N_i G_i^j, 1 \leq j \leq n_{MTC} \quad (9)$$

A. Hierarchical Clustering for Discrete-Continuous Data

The data for each patient contains injury severity, both land and air travel time to all the centers, and air travel time from the helicopter depots to the patient, which means that the data is a mixture of binary and real-valued numbers. As Section III-A shows, the assigned center depends on the injury and location of the patient. The injury severity is a binary value (triaged to MTC or TU), and the location can be presented by the land travel time to all the centers, which is a real vector.

To handle the discrete-continuous data, a hierarchy clustering technique [76] is adopted in this work. In the first hierarchy clustering, only the location of patients are considered. The dimension of the vector of land travel time equals to the number of centers, which is much higher than the dimension of a location. Therefore, principal components analysis (PCA) [77] is employed to reduce the dimension. Then, patients are divided into two classes inside each cluster in the first hierarchy according to the severity of injuries, which forms the second hierarchy of clusters. Finally, the center of every cluster is used as the representative patient.

Fig. 8 is an example with 12 patients to illustrate the hierarchical clustering. We first divide the data into K clusters based on the location, where $K = 2$ in the example. Then, a further classification is applied inside the K clusters based on the injury severity, i.e. patients triaged to MTC are classified into one sub-cluster, and those triaged to TU are classified into another sub-cluster. The center of every sub-cluster, represented by a pentagram in Fig. 8, will be used as one representative patient in the approximated objective and constraint evaluations using Equations (6)-(9).

B. Analysis of Surrogates with Various Fidelities

As mentioned above, the distribution of the data is naturally shows particular patterns, which makes it possible to group the data into a number of clusters. Intuitively, the cluster number, K , influences not only the fidelity of the estimated objectives and constraints, but the computational expense as well. In order to better understand the influence of K on environmental selection and the accuracy of the estimated objective and constraint, we generate a reference set by running NSGA-II using exact function evaluations (i.e., using all patients in the data) using Equations (1) - (4) for 100 generations, where the population size is set to 100.

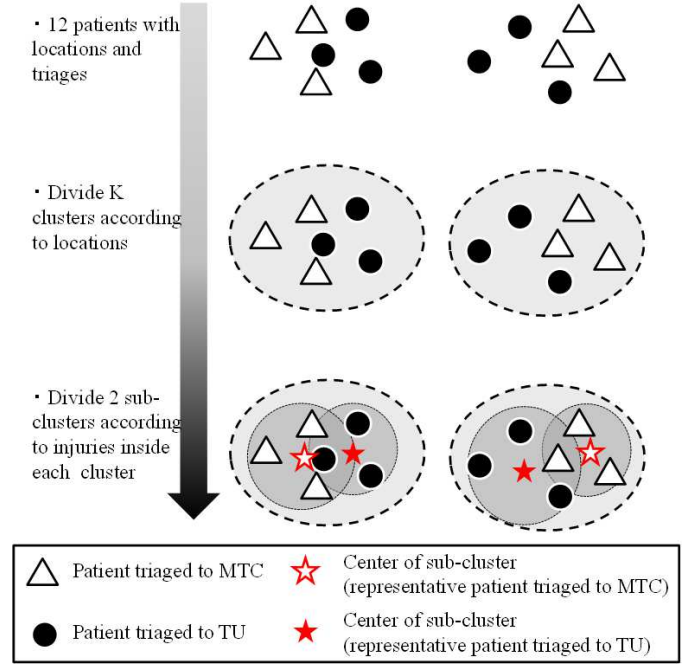


Fig. 8. An illustrative example of hierarchical clustering.

To examine the influence of the cluster number on selection, we compare the parents selected according to the objectives and constraints calculated using Equations (6) to (9) at each generation with those selected in the reference set and calculate the percentage of the correctly selected solutions out of 100 selected parents. This percentage is averaged over 100 generations and is used as an indicator for accuracy of selection. The change of the averaged accuracy of selection over the number of clusters is plotted in the left panel of Fig. 9.

Similarly, we calculate the mean absolute percentage error (MAPE) averaged over 200 individuals (the combined population before selection) at each generation for the two objectives and two constraints, respectively. Finally, these errors are again averaged over 100 generations and plotted in the right panel in Fig. 9. Since the constraint on TU proximation is not data-driven, it is not considered here.

From the results in Fig. 9, we can see that as K increases, the average MAPE of the objectives and constraints decrease rapidly. Accordingly, the accuracy of selection increases. When K reaches 2000, the approximation errors do not decrease significantly and the selection accuracy no longer improves. This indicates that the evolutionary algorithm will not benefit further when the number of clusters is larger than 2000, although the cost of computational expense. Thus, the key issue in saving computation time without sacrificing too much on the solution quality will be to properly tune the number of clusters so that the evolutionary algorithm is still able to find the optimal solutions at a much lower computational cost.

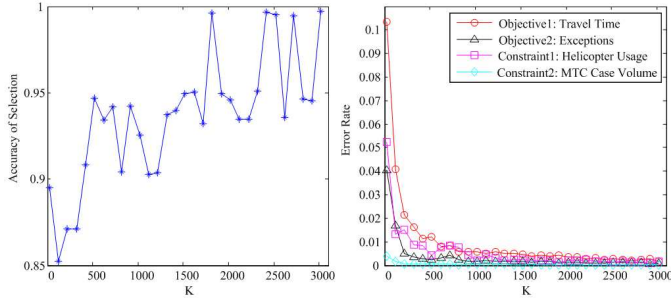


Fig. 9. Accuracy of selection and approximation error of the objectives and constraints over the change of the number of clusters K s.

C. Management of Multi-Fidelity Surrogates

The surrogate management by adjusting the fidelity of the estimated objective and constraints by tuning the number of clusters K . One intuitive assumption here is the the approximation is good enough if the estimated objective functions can ensure that the same individuals are selected as the exact objective functions. For this reason, let us take a look at the relationship between the approximation error on the two objectives and sorted no-dominated fronts. Fig. 10 shows the solutions in the first front (denoted by circles) and the last front (denoted by dots) that will be selected as parents based on the exact function evaluations. Now if a solution on the first front is evaluated using the approximated objectives and the maximum absolute approximation errors on the two objectives are ER_1 and ER_2 , respectively. As a result, the location of this solution evaluated based on the clustered data using Equations in (6) to (9) may locate anywhere in the shaded box defined by $2ER_1 \times 2ER_2$. When there is no overlap between two grey rectangles of the solutions from the first and last fronts, the selection will not be affected by the approximated errors. Therefore, the acceptable maximum error on the two objectives, ER_i^* , $i = 1, 2$, can be calculated by Equation (10), where F_1 is the set of solutions on the first front and F_l is the set of solutions on the last front that will be selected as parent solutions for the next generation.

$$ER_i^* = \frac{1}{2} \min\{f_i^k - f_i^j\}, 1 \leq k \leq |F_l|, 1 \leq j \leq |F_1| \quad (10)$$

Given the acceptable maximum errors ER_1^* and ER_2^* , the desired minimum number of clusters K can be determined if the relationship between the maximum acceptable errors and the minimum K can be obtained. To achieve this, we construct a regression model with the historical data pairs between the cluster number and the approximation errors. For the trauma system design problem, f_1 is continuous and f_2 is discrete, and the errors on f_1 are usually larger than that on f_2 , as shown in Fig. 10. For simplicity, we choose to use the errors on f_1 to estimate the needed number of clusters. In addition, as observed in Fig. 9, the relationship between the error and the number of clusters is nonlinear. Therefore, we use the following model to estimate the error from a given K :

$$ER = \frac{1}{\beta_1 + \beta_2 K}, \quad (11)$$

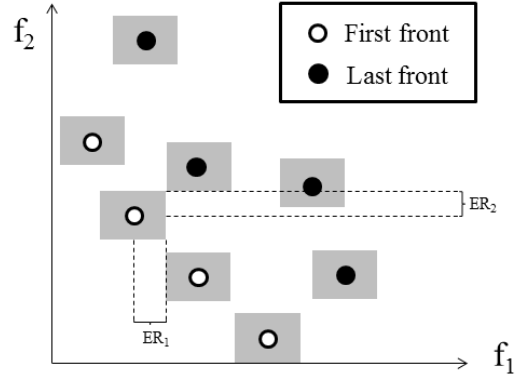


Fig. 10. Influence of the approximation errors on non-dominated sorting in a population.

where β_1 and β_2 are two parameters, K is re-set as $\frac{1}{\beta_2}(\frac{1}{ER^*} - \beta_1)$.

Errors can be evaluated by comparing the approximated objective values with the exact evaluations. To reduce computational cost, we only evaluate the non-dominated solutions at each generation using exact function evaluations and calculate the approximation errors on the first objective. The maximum error at each generation is recorded as ER together with the cluster number the to be used for estimate the parameters in the regression model in Equation (11).

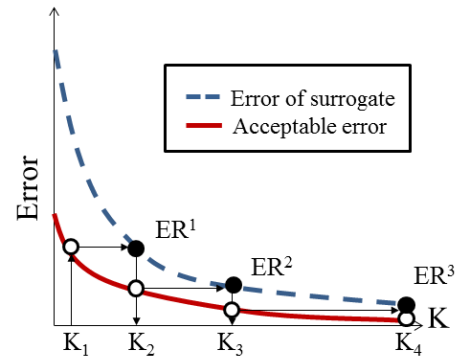


Fig. 11. An example of the surrogate management, where K is changed from K_1 to K_4 . When the approximation error for the current K is larger than the acceptable error, a new K will be calculated according to the regression model in an effort to reduce the approximation error to an accepted level.

Fig. 11 illustrates one scenario of the proposed surrogate management strategy based on the regression model. Given the current cluster number K_1 , the maximum error caused by the approximate functions is larger than the acceptable error ER^1 , which means that K_1 is too small and should be increased. Based on the regression model, a new cluster number, K_2 is obtained. For K_2 , the new approximation error is calculated, which is still larger than the acceptable error ER^2 . This procedure continues until a cluster number K_4 is adopted, where results in an approximation error smaller than the acceptable error. Therefore, K_4 will be adopted as the cluster number for further evolutionary optimization.

The main steps of the algorithm for adapting K are pre-

Algorithm 1 Adaptive multi-fidelity surrogate using a regression model.

Input: K -the number of clusters, K_{max} -maximal K , ER^* -the acceptable error of f_1 , S -historical pairs of (K, ER) .

Output: K_n .

```

1: if no improvement in the non-dominated set then
2:   Evaluate the non-dominated solutions using all records
     (exact evaluations).
3:   Calculate  $ER$ .
4:   Add the pair  $(K, ER)$  to  $S$ .
5:   if  $|S| < 2$  then
6:      $K_n = 2K$ .
7:   else
8:     Regression on  $S$  by Equation (11).
9:     Calculate  $ER^*$  by Equation (10).
10:    if  $ER^* < ER/2$  then
11:       $ER^* = ER/2$ .
12:    end if
13:     $K_n = \frac{1}{\beta_2}(\frac{1}{ER^*} - \beta_1)$ .
14:  end if
15:  if  $K_n > K_{max}$  then
16:     $K_n = K_{max}$ .
17:  end if
18: else
19:    $K_n = K$ .
20: end if

```

sented in Algorithm 1. In the algorithm, K starts with a small number, typically the number of trauma centers. For a given K , the evolutionary optimization will be run for a number of generations no improvement can be achieved. This is realised by comparing the obtained non-dominated sets between consecutive generations. When there is no improvement, the algorithm considers the current K too small and then adapt the number K using regression model.

A few special situations need to be taken into account. For example, if the historical data pairs (K, ER) are insufficient for building the regression model, K is simply doubled to add a new data pair. In addition, dramatic increase in K should be avoided, which is realized by limiting the maximum decrease in the acceptable error to the half of the current error, as described in lines 10-12 in Algorithm 1. Finally, we set the maximum of K to be K_{max} to limit the computational cost.

V. EXPERIMENTAL RESULTS

A. Experimental Settings

To evaluate effectiveness of the proposed multi-fidelity surrogate management strategy, we compare the performance of three variants of the NSGA-II for the trauma system design, i.e., NSGA-II assisted with the adaptive multi-fidelity surrogate management strategy, NSGA-II with a fixed cluster number with K being fixed to 20, and NSGA-II using exact function evaluations. The population size of the three compared algorithms is set to 100 and the maximum of generations is 200. 20 independent runs are performed for the two variants using clustered data while five independent

runs are performed for the NSGA-II using exact functions evaluations. Note that each run of the NSGA-II using exact function evaluations (i.e., using all 40,000 records) takes about 45 hours. The non-dominated solution sets obtained from the five independent runs of the NSGA-II using exact fitness evaluations are combined and the non-dominated solutions of these combined solutions are used as the reference set for calculating performance indicators such as the inverted generational distance (IGD) [78] or purity [79]. Note that we normalize the objectives by the extreme solutions in the reference set because of the different scale of two objectives. Furthermore, we use the number of calls to the patient allocation algorithm to evaluate the computational expense. For each exact function evaluations, the number of calls equals to the number of patients, while for clustered data, the number of calls equals to the number of data clusters.

B. Non-Adaptive Surrogate Management

We first examine the performance of the NSGA-II using clustered data with the cluster number K being fixed during the evolutionary optimization. To investigate the influence of the cluster number K on the performance, K has been set to ?? For each fixed K , 20 independent runs are performed. For The average IGD value and the number of allocated patients over the change of K are shown in Fig. 12.

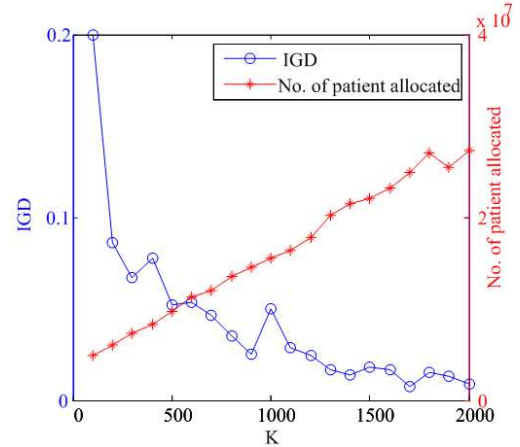


Fig. 12. Average IGD values and the number of allocated patients for different K s.

From Fig. 12, we can see that the number of allocated patients linearly increases with the growing K , which represents the computation time. As K increases, more patients (represented by the cluster centers) need to be allocated in objective evaluations, leading to increases in the computation time. At the same time, the approximation accuracy of the surrogates as the cluster number increases resulting in a better IGD. However, the improvement of the IGD is minor when K is larger than 1500 in comparison to the increases in the computational cost.

C. Adaptive Surrogate Management

1) *Maximal Number of Clusters:* For the surrogate management algorithm with an adaptive cluster number, the lower and

upper bounds of K need to be predefined. As we mentioned before, the lower bound of K is usually set to the number of trauma centers (18 in this example), while the upper bound K_{max} is a parameter to be specified. In the following, we first test the sensitivity of the performance to different specifications of K_{max} .

For this purpose, we run the proposed algorithm with K_{max} being set to 500, 1000, 1500, and 2000, respectively, and each case is run for 20 independent times. The trade-off between the performance and computation time is plotted in Fig. 13, where purity (the rate of non-dominated solutions in the union set of the compared algorithms) is used as the performance indicator.

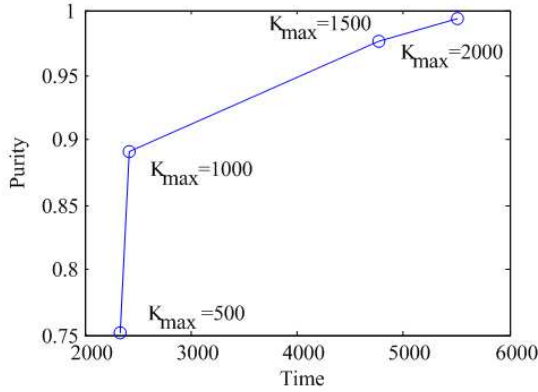


Fig. 13. Trade-off between the performance in terms of purity and computation time in the the proposed multi-fidelity surrogate management scheme.

From Fig. 13, it is clear that a larger K_{max} can bring about better performance but also increases computation time. Interestingly, we notice there is a knee point in the trade-off between purity and computational cost when $K_{max} = 1000$.

2) *Adaptive Number of Clusters Based on Regression:* To better understand the behavior the proposed algorithm, we perform additional runs of the proposed algorithm with $K_{max} = 1000$ for 20 independent times in order to take a look into the details in adaptive surrogate management, including the changes of the acceptable error, change of the parameters in the regression model, and the change of clusters K during the whole evolutionary optimization.

The average ER (the real approximation error of the surrogate) and ER^* (the maximum acceptable error) as K adapts are presented in Fig. 14. We can see that both ER and ER^* decrease as K increases, which is very similar to the changes shown in Fig. 9. However, the decrease of ER^* is not as significant ER . Another observation we can make is that at the end of the evolutionary run, the difference between ER and ER^* becomes very small as K approaches to $K_{max} = 1000$.

To check whether the estimation of the parameters in the regression model described in Equation (11) converge, Fig. 15 records the change of the parameters β_1 and β_2 over the change of K during the evolutionary optimization. The results are averaged over 20 runs. From these results we can see that the estimation of the parameters in the regression model is able to converge.

Furthermore, we record in Fig. 16 the changes of K as the evolution proceeds and compare the difference in the change

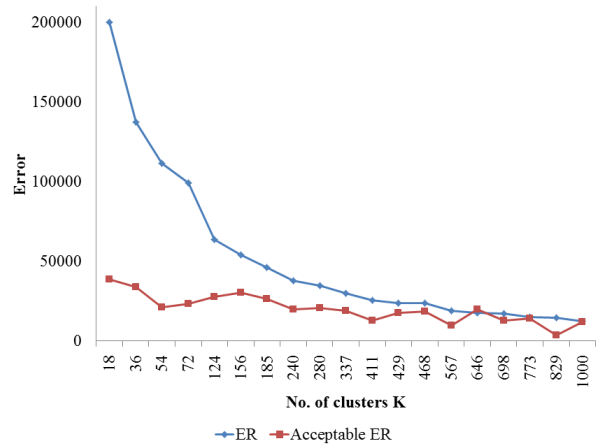


Fig. 14. Average ER and ER^* over different K s.

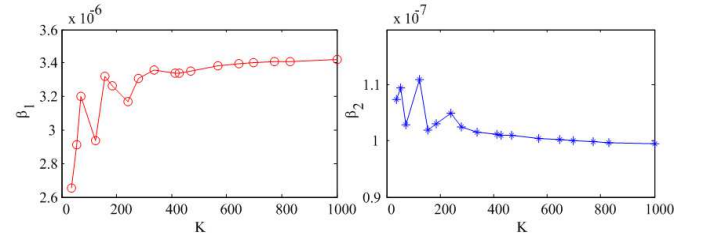


Fig. 15. The convergence profile of β_1 and β_2 as the cluster number K increases during the evolutionary optimization. Results are averaged over 20 runs.

of K with and without the smoothness control strategy as implemented in lines 10-12 in Algorithm 1. From these results can see that without the smoothness control, K increased dramatically in between and altogether 11 changes are needed, whereas with the change smoothness control, K increases more smoothly and only 10 changes are needed. We will show in the next section that the smoothness control in change of K also results in slightly better performance.

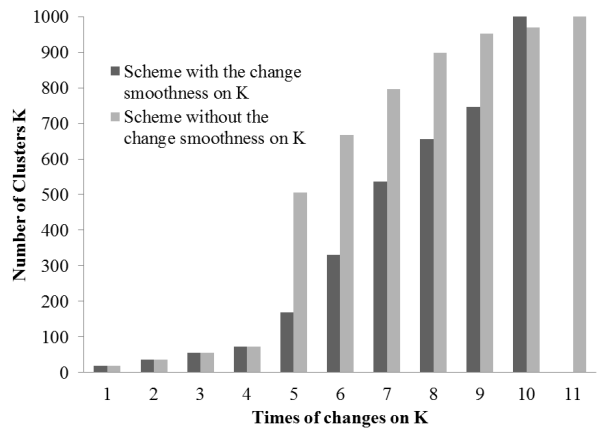


Fig. 16. Changes of K in the adaptive surrogate management scheme with and without the change smoothness.

D. Comparison of Computational Cost

In this subsection, we compare the computational cost of the NSGA-II using proposed adaptive surrogate management scheme with the NSGA-II using exact function evaluations (NSGA-II for short). Here, we also consider two minor variants of the proposed algorithm, one with smoothness control in change of K , termed SA-NSGA-II with CS, one without, termed SA-NSGA-II without CS. The three compared algorithms are all tun for 100 generations, however, NSGA-II is repeated for five times due to its high computational cost, while the two variants of SA-NSGA-II are repeated for 20 times. Here, we use the non-dominated set of the combined results from the five runs of the NSGA-II as the reference set for calculating the IGD. The IGD values and computation times (in seconds) of these three algorithms are presented in Table II. From these results, the IGD value of both SA-NSGA-II variants is very close to 0, which means that the solution sets they obtained are very similar to the reference set. However, the computation time needed for the two variants of SA-NSGA-II is much shorter than NSGA-II and with the smoothness control of changes in K , the computation time is further reduced with even slightly better performance.

TABLE II
COMPARISON OF PERFORMANCE AND COMPUTATIONAL COST

	IGD	Time (s)
SA-NSGA-II with CS	$2.47\text{e-}02 \pm 1.96\text{e-}02$	$6.54\text{e}+03 \pm 3.57\text{e}+03$
SA-NSGA-II without CS	$2.51\text{e-}02 \pm 2.79\text{e-}02$	$10.31\text{e}+03 \pm 5.00\text{e}+03$
NSGA-II	$0.00\text{e}+00 \pm 0.00\text{e}+00$	$8.14\text{e}+04 \pm 5.36\text{e}+02$

While the above comparison is based on the same number of generations, it is also of interest to compare the performance of the algorithms using the computation time as the baseline. For this purpose, we set the stopping criterion to be 1 hour for each of the three compared and then calculate the IDG value of the solution set using the same reference set as the above experiment. The results averaged over 20 independent runs are presented in Table III. By using the Wilcoxon signed-rank test [80], it has been found that the IGD values of the two SA-NSGA-II variants are significantly better than that of the NSGA-II. In addition, we can see that the IGD value of SA-NSGA-II with CS is again better than SA-NSGA-II without CS control. The non-dominated solution sets obtained the three algorithms in the run with the median IGD value are plotted in Fig. 17.

TABLE III
THE IGD VALUES OF THE SA-NSGA-II VARIANTS AND NSGA-II USING 1 HOUR RUNTIME AS THE STOPPING CRITERION.

	IGD
SA-NSGA-II with CS	$2.69\text{e-}02 \pm 1.93\text{e-}02$
SA-NSGA-II without CS	$4.31\text{e-}02 \pm 1.01\text{e-}01$
NSGA-II	$9.57\text{e-}02 \pm 5.08\text{e-}02$

It is of interest to take a look at a few designs selected from the non-dominated solution set to understand the optimal designs obtained by SA-NSGA-II with CS. The resulting configurations of the four solutions, as indicated by A, B, C, and D in Fig. 17, are shown in the map in Fig. 18. It is clear

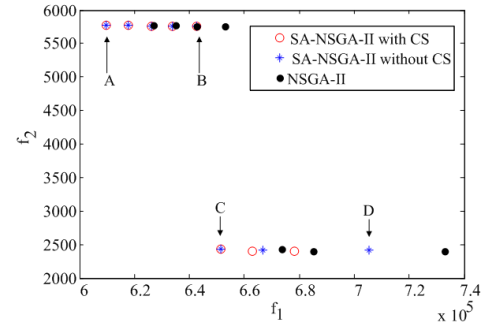


Fig. 17. Non-dominated solution sets of the SA-NSGA-II variants and NSGA-II obtained by stopping the algorithms after 1 hour.

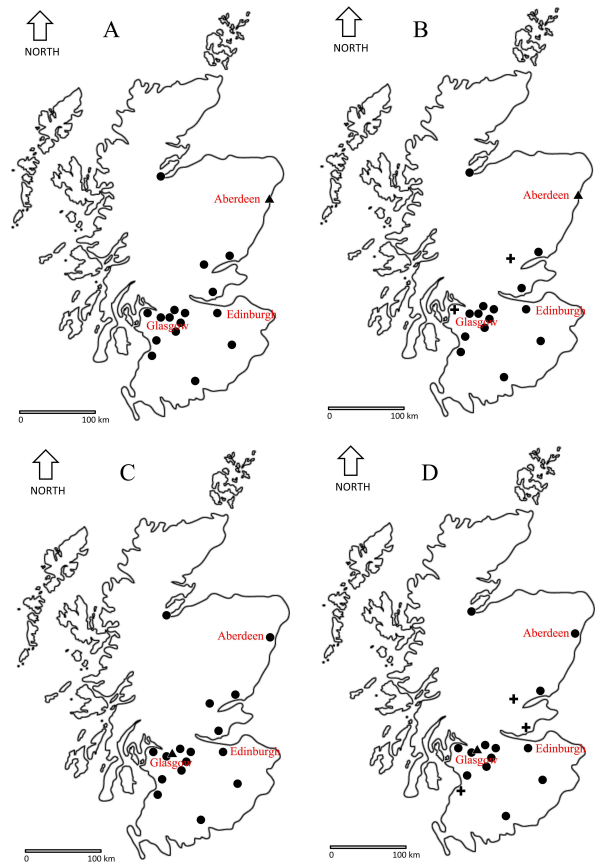


Fig. 18. Different solutions (A-D) on the map, where triangles are MTCs, circles are TUs, and plus signs are LEHs.

that those solutions can be classified into two categories, where in solutions A and B one MTC is established in Aberdeen, and in solutions C and D one MTC in Glasgow. These two different choices for MTC actually match the patient density in Fig. 7, where Aberdeen and Glasgow are two centers of high incidence areas. Since Aberdeen is located closer to the middle of Scotland than Glasgow, the total travelling time of solutions A and B is less than that of C and D. However, in A and B, patients triaged to MTC in the south of Scotland cannot be timely sent to the MTC in Aberdeen, which increases the number of MTC exceptions. Therefore, C and D are better than A and B in terms of MTC exceptions. Compared with A and

C, B and D have a smaller number of TUs, which significantly increases the traveling time. It is also noticed that the obtained solutions requires only one MTC, which is smaller than the current configuration in Scotland, which has four MTCs. **How to compare the performance of the obtained solutions with the existing one?**

VI. CONCLUDING REMARKS

This paper discuss in detail data-driven optimization problems and categorizes them into two large groups, one termed off-line data-driven optimization problems where no incremental data is available during optimization, and the other on-line data-driven optimization problems where new data can be collected during the optimization. The latter can further to divided into two types of problems, one of which can control the position of the newly sampled data, while the other cannot. Main challenges in data-drive evolutionary optimization, in addition to the high cost in collecting the data, consists in the quantity, quality and other nature of the data. As data-driven evolutionary optimization typically needs to be assisted by surrogates, most challenges in surrogate-assisted evolutionary optimization will also need to be addressed. However, not much work in surrogate-assisted evolutionary optimization has considered difficulties in the presence of huge amount of data, or imbalanced, ill-distributed, or heterogeneous data.

In this work, we concentrate on addressing the challenge in evolutionary of trauma systems, which is a real-world off-line data-driven evolutionary optimization that involves in large amount of data. The main idea is to group the data into a number of clusters using a hierarchical clustering algorithm, thereby reducing the amount of computation time. To achieve an optimal balance between computation time and the solution quality, we propose a surrogate management scheme by establishing a regression model that can estimate the needed number of clusters based on the maximum acceptable approximation error. The experimental results demonstrate that the proposed algorithm is able to considerably reduce computational time of the data-driven evolutionary algorithm using all data for function evaluations.

Although the proposed multi-fidelity surrogate management strategy is specifically developed for design of trauma systems, the main ideas in this algorithm are likely to be applied to other data-driven optimization problems relying on huge amount of data for function evaluations. However, the clustering algorithm used and the regression model for estimating the needed number of clusters may need to be tailored for the problems to be solves. Future efforts will also be dedicated to addressing the other challenges imposed by the particular nature of the data, e.g., by combining advanced learning and evolutionary techniques for dealing with big data [81], [53].

REFERENCES

- [1] P. J. Fleming and R. C. Purshouse, "Evolutionary algorithms in control systems engineering: a survey," *Control engineering practice*, vol. 10, no. 11, pp. 1223–1241, 2002.
- [2] D. Dasgupta and Z. Michalewicz, *Evolutionary algorithms in engineering applications*. Springer Science & Business Media, 2013.
- [3] K. De Grave, J. Ramon, and L. De Raedt, "Active learning for high throughput screening," in *Discovery Science*. Springer, 2008, pp. 185–196.
- [4] K. Giannakoglou, "Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence," *Progress in Aerospace Sciences*, vol. 38, no. 1, pp. 43–76, 2002.
- [5] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE Computational Intelligence Magazine*, vol. 4, no. 3, pp. 62–76, 2009.
- [6] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.
- [7] A. Eiben and J. Smith, "Interactive evolutionary algorithms," in *Introduction to Evolutionary Computing*. Springer, 2015, pp. 215–222.
- [8] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.
- [9] J. O. Jansen, J. J. Morrison, H. Wang, R. Lawrenson, G. Egan, S. He, and M. K. Campbell, "Optimizing trauma system design: the geos (geospatial evaluation of systems of trauma care) approach," *Journal of Trauma and Acute Care Surgery*, vol. 76, no. 4, pp. 1035–1040, 2014.
- [10] J. O. Jansen, J. J. Morrison, H. Wang, S. He, R. Lawrenson, J. Hutchison, and M. Campbell, "Access to specialist care: Optimizing the geographic configuration of trauma systems," *Journal of Trauma and Acute Care Surgery*, vol. 00, no. 0, pp. 1–10, 2015, doi=10.1097/TA.0000000000000827.
- [11] K. Miettinen, *Nonlinear multiobjective optimization*. Springer, 1999.
- [12] A. Zhou, B. Qu, H. Li, S. Zhao, P. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.
- [13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.
- [14] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 4, pp. 257–271, 1999.
- [15] H. Wang and X. Yao, "Corner sort for Pareto-based many-objective optimization," *Cybernetics, IEEE Transactions on*, vol. 44, no. 1, pp. 92–102, 2014.
- [16] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, 2007.
- [17] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 6, pp. 909–923, 2014.
- [18] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature-PPSN VIII*. Springer, 2004, pp. 832–842.
- [19] J. O. Jansen, J. J. Morrison, H. Wang, S. He, R. Lawrenson, M. K. Campbell, and D. R. Green, "Feasibility and utility of population-level geospatial injury profiling: prospective, national cohort study," *Journal of Trauma and Acute Care Surgery*, vol. 78, no. 5, pp. 962–969, 2015.
- [20] M. Tabatabaei, J. Hakanen, M. Hartikainen, K. Miettinen, and K. Sindhya, "A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods," *Structural and Multidisciplinary Optimization*, vol. 52, no. 1, pp. 1–25, 2015.
- [21] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing*, vol. 9, no. 1, pp. 3–12, 2005.
- [22] J. Knowles and H. Nakayama, "Meta-modeling in multiobjective optimization," in *Multiobjective Optimization*. Springer, 2008, pp. 245–284.
- [23] I. Loshchilov, M. Schoenauer, and M. Sebag, "A mono surrogate for multiobjective optimization," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM, 2010, pp. 471–478.
- [24] Y. Jin and B. Sendhoff, "Reducing fitness evaluations using clustering techniques and neural network ensembles," in *Proceedings of the 6th annual conference on Genetic and evolutionary computation*. Springer, 2004, pp. 688–699.
- [25] D. Lim, Y.-S. Ong, Y. Jin, and B. Sendhoff, "A study on metamodeling techniques, ensembles, and multi-surrogates in evolutionary computation," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM, 2007, pp. 1288–1295.
- [26] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, 2010.

- [27] E. Rigoni and A. Turco, "Metamodels for fast multi-objective optimization: trading off global exploration and local exploitation," in *Simulated Evolution and Learning*. Springer, 2010, pp. 523–532.
- [28] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum, "Combining global and local surrogate models to accelerate evolutionary optimization," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 1, pp. 66–76, 2007.
- [29] Y. Tenne and S. W. Armfield, "A framework for memetic optimization using variable global and local surrogate models," *Soft Computing*, vol. 13, no. 8–9, pp. 781–793, 2009.
- [30] Z. Zhou, Y. S. Ong, M. H. Nguyen, and D. Lim, "A study on polynomial regression and gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm," in *Evolutionary Computation, 2005. CEC'05. The 2003 Congress on*, vol. 3. IEEE, 2005, pp. 2832–2839.
- [31] D. Büche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with gaussian process fitness function models," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 35, no. 2, pp. 183–194, 2005.
- [32] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using gaussian process-based inverse modeling," *Evolutionary Computation, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015, doi=10.1109/TEVC.2015.2395073.
- [33] M. Emmerich, "Single-and multi-objective evolutionary design optimization assisted by Gaussian random field metamodels," Ph.D. dissertation, 2005.
- [34] B. Liu, Q. Zhang, and G. G. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 2, pp. 180–192, 2014.
- [35] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with gaussian process model," *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 3, pp. 456–474, 2010.
- [36] I. Loshchilov, M. Schoenauer, and M. Sebag, "Comparison-based optimizers need comparison-based surrogates," in *Parallel Problem Solving from Nature, PPSN XI*. Springer, 2010, pp. 364–373.
- [37] L. Bajer and M. Holeňa, "Surrogate model for continuous and discrete genetic optimization based on rbf networks," in *Intelligent Data Engineering and Automated Learning—IDEAL 2010*. Springer, 2010, pp. 251–258.
- [38] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 3, pp. 326–347, 2014.
- [39] S. Zapotecas Martínez and C. A. Coello Coello, "Moea/d assisted by rbf networks for expensive multi-objective optimization problems," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. ACM, 2013, pp. 1405–1412.
- [40] N. Azzouz, S. Bechikh, and L. Ben Said, "Steady state ibea assisted by mlp neural networks for expensive multi-objective optimization problems," in *Proceedings of the 16th annual conference on Genetic and evolutionary computation*. ACM, 2014, pp. 581–588.
- [41] A. E. Brownlee, J. McCall, Q. Zhang *et al.*, "Fitness modeling with markov networks," *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 6, pp. 862–879, 2013.
- [42] A. Gaspar-Cunha and A. Vieira, "A multi-objective evolutionary algorithm using neural networks to approximate fitness evaluations," *Int. J. Comput. Syst. Signal*, vol. 6, no. 1, pp. 18–36, 2005.
- [43] A. Giotis, K. Giannakoglou, and J. Périaux, "A reduced-cost multi-objective optimization method based on the pareto front technique, neural networks and pvm," in *Proceedings of the ECCOMAS*, 2000.
- [44] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 5, pp. 481–494, 2002.
- [45] —, "On evolutionary optimization with approximate fitness functions," in *Proceedings of the 2nd annual conference on Genetic and evolutionary computation*, 2000, pp. 786–793.
- [46] L. Gräning, Y. Jin, and B. Sendhoff, "Efficient evolutionary optimization using individual-based evolution control and neural networks: A comparative study," in *European Symposium on Artificial Neural Networks*, 2005, pp. 273–278.
- [47] H. Ulmer, F. Streichert, and A. Zell, "Evolution strategies assisted by gaussian processes with improved preselection criterion," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, vol. 1. IEEE, 2003, pp. 692–699.
- [48] J. Branke and C. Schmidt, "Faster convergence by means of fitness estimation," *Soft Computing*, vol. 9, no. 1, pp. 13–20, 2005.
- [49] Y. Jin, M. Hüsken, and B. Sendhoff, "Quality measures for approximate models in evolutionary computation," in *Proceedings of the 5th annual conference on Genetic and evolutionary computation*, 2003, pp. 170–173.
- [50] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 3, pp. 329–355, 2010.
- [51] M. Hüsken, Y. Jin, and B. Sendhoff, "Structure optimization of neural networks for evolutionary design optimization," *Soft Computing*, vol. 9, no. 1, pp. 21–28, 2005.
- [52] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. Coello Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part I," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 1, pp. 4–19, 2014.
- [53] S. Thomas and Y. Jin, "Reconstructing gene regulatory networks: Where optimization meets big data," *Evolutionary Intelligence*, vol. 7, no. 1, pp. 29–47, 2014.
- [54] S. Wang and X. Yao, "Using class imbalance learning for software defect prediction," *Reliability, IEEE Transactions on*, vol. 62, no. 2, pp. 434–443, 2013.
- [55] J. L. Arbuckle, G. A. Marcoulides, and R. E. Schumacker, "Full information estimation in the presence of incomplete data," *Advanced structural equation modeling: Issues and techniques*, vol. 243, p. 277, 1996.
- [56] Y. Liu, F. Shang, L. Jiao, J. Cheng, and H. Cheng, "Trace norm regularized candecomp/parafac decomposition with missing data," *Cybernetics, IEEE Transactions on*, vol. 45, no. 11, pp. 2437–2448, 2015.
- [57] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.
- [58] H. Wang, Q. Zhang, L. Jiao, and X. Yao, "Regularity model for noisy multiobjective optimization," *Cybernetics, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015, doi=10.1109/TCYB.2015.2459137.
- [59] Y. Jin, K. Tang, X. Yu, B. Sendhoff, and X. Yao, "A framework for finding robust optimal solutions over time," *Memetic Computing*, vol. 5, no. 1, pp. 3–18, 2013.
- [60] R. R. Chan and S. D. Sudhoff, "An evolutionary computing approach to robust design in the presence of uncertainties," *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 6, pp. 900–912, 2010.
- [61] S. Castano and V. De Antonellis, "Global viewing of heterogeneous data sources," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 13, no. 2, pp. 277–297, 2001.
- [62] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 1, pp. 97–107, 2014.
- [63] P. Arabie, L. J. Hubert, and G. De Soete, *Clustering and classification*. World Scientific, 1996.
- [64] J. Luo, L. Jiao, and J. Lozano, "A sparse spectral clustering framework via multi-objective evolutionary algorithm," *Evolutionary Computation, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015, doi=10.1109/TEVC.2015.2476359.
- [65] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
- [66] F. Mosteller and J. W. Tukey, "Data analysis and regression: a second course in statistics," *Addison-Wesley Series in Behavioral Science: Quantitative Methods*, 1977.
- [67] C. Mock, *Guidelines for essential trauma care*. World Health Organization, 2004.
- [68] B. J. Gabbe, G. D. Biostat, P. M. Simpson, A. M. Sutherland, G. Dip, R. Wolfe, M. C. Fitzgerald, R. Judson, and P. A. Cameron, "Improved functional outcomes for major trauma patients in a regionalized, inclusive trauma system," *Annals of surgery*, vol. 255, no. 6, pp. 1009–1015, 2012.
- [69] E. J. MacKenzie, S. Weir, F. P. Rivara, G. J. Jurkovich, A. B. Nathens, W. Wang, D. O. Scharfstein, and D. S. Salkever, "The value of trauma center care," *Journal of Trauma-Injury, Infection, and Critical Care*, vol. 69, no. 1, pp. 1–10, 2010.
- [70] A. C. of Surgeons Committee on Trauma *et al.*, "Resources for optimal care of the trauma patient," *American College of Surgeons, Chicago, IL*, 2006.
- [71] J. O. Jansen and M. K. Campbell, "The geos study: Designing a geospatially optimised trauma system for scotland," *The Surgeon*, vol. 12, no. 2, pp. 61–63, 2014.
- [72] C. A. Macias, M. R. Rosengart, J.-C. Puyana, W. T. Linde-Zwirble, W. Smith, A. B. Peitzman, and D. C. Angus, "The effects of trauma

- center care, admission volume, and surgical volume on paralysis following traumatic spinal cord injury,” *Annals of surgery*, vol. 249, no. 1, p. 10, 2009.
- [73] N. Mladenović and P. Hansen, “Variable neighborhood search,” *Computers & Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997.
 - [74] P. Hansen and N. Mladenović, “Variable neighborhood search: Principles and applications,” *European journal of operational research*, vol. 130, no. 3, pp. 449–467, 2001.
 - [75] E. K. Burke, A. J. Eckersley, B. McCollum, S. Petrovic, and R. Qu, “Hybrid variable neighbourhood approaches to university exam timetabling,” *European Journal of Operational Research*, vol. 206, pp. 46–53, 2010.
 - [76] G. W. Milligan and M. C. Cooper, “A study of the comparability of external criteria for hierarchical cluster analysis,” *Multivariate Behavioral Research*, vol. 21, no. 4, pp. 441–458, 1986.
 - [77] G. H. Dunteman, *Principal components analysis*. Sage, 1989, no. 69.
 - [78] Q. Zhang, A. Zhou, S. Zhao, P. Suganthan, W. Liu, and S. Tiwari, “Multiobjective optimization test instances for the CEC 2009 special session and competition,” *University of Essex, Colchester, UK and Nanyang Technological University, Singapore, Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report*, pp. 1–30, 2008.
 - [79] S. Bandyopadhyay, S. Pal, and B. Aruna, “Multiobjective GAs, quantitative indices, and pattern classification,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 5, pp. 2088–2099, 2004.
 - [80] M. Hollander and D. Wolfe, *Nonparametric statistical methods*. Wiley-Interscience, 1999.
 - [81] Z.-H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams, “Big data opportunities and challenges: Discussions from data analytics perspectives,” *IEEE Computational Intelligence Magazine*, vol. 9, no. 4, pp. 62–74, 2014.