The Case Ordering Problem in Surgical Procedural Training through Process Mining

Felipe Cornejo*, Cristian Fazio*, Jorge Munoz-Gama*, Marcos Sepúlveda*,
Ricardo Fuentes† and Rene de la Fuente†
*Department of Computer Science, School of Engineering
†Department of Anesthesiology, School of Medicine
Pontificia Universidad Católica de Chile (Chile)

***Abstract*—Recently, process perspective has been incorporated into the surgical training. These approaches use Process Mining to provide instructors with procedural reports of each student. However, reviewing those reports in a random order could be unnecessarily tedious: each report could be totally different from the previous one, preventing the instructors of any comparative aid, and forcing them to change their mindset every time. Complementary to other approaches such as clustering or tacit knowledge, we propose an approach to optimize, from a process perspective, the ordering of the students executions, such that the procedural distance between an execution and the next one is the lowest possible. The contribution of this paper is twofold: first, the problem is formalized; second, a method for solving the problem tailored to surgical training is presented. Preliminary results are provided, and the preliminary validity of the approach has been corroborated by a medical expert.**

***Index Terms*—process mining, case ordering, healthcare, medical training, surgical procedures**

I. Introduction

The usage of the process perspective in the healthcare analysis is not new. In [1] the authors presented a literature review, identifying more than 74 study cases where Process Mining [2] was used to analyze the process perspective, in fields such as dentistry [3], pediatrics [4], and oncology [5]. Surgery is no exception to the analysis of this process perspective, e.g., in [6], [7], authors propose models and frameworks to aid on the analysis of surgical procedures.

Recently, a new research line has appeared, where the process perspective is analyzed in the context of medical education, and specifically on the training of surgical procedural skills. In [8], [9] the authors propose a novel method where student training executions are recorded to be used as valid observational data [7]. Activities observed in the recordings are tagged, and Process Mining is then used on this event data to generate procedural reports for the students [8], [9]. Those reports provide end-to-end process models of the execution, and complement other existing feedback/evaluation instruments such as Checklists and Global Rating Scales [10]. A partial example of such reports from [9] is shown in Figure 1.

This novel method has resulted in new set of challenges. For instance, from the course instructors perspective, reviewing a procedural report of one of their students is a non trivial task. Focus and accuracy is required to understand the end-to-end process models. If the instructor needs to review n reports at a sitting, and the reports are not ordered in any specific order, that could be an unnecessary tedious task, i.e., each report may be totally different from the previous one, so the instructor needs to “reset” his/her mindset, and start the analysis from scratch. Several alternatives could be applied to mitigate this problem. For instance, clustering techniques [11] could be apply to group the n reports into clusters of similar reports. Other techniques, such as tacit knowledge based approaches [12], instructors reviews against their knowledge about how the procedure must be executed.

This work explores a complementary approach to order the reports, where consecutive reports are ordered by similarity. The driven idea is that the mindset change that would require to change from one report to a similar next one would be lower, and consequently the instructor would be able to use the recent effort as a prior for the new analysis, improving the efficiency and reducing the tediousness. Notice that this new approach could be complementary to other approaches, e.g., a large set of reports are clustered, and within each group the reports are ordered by consecutive similarity.

This paper proposes a method to provide an ordering of surgical procedure training course executions in which the sum of procedural difference between consecutive executions is the lowest possible. The contribution of this paper is twofold: first, the *Optimal Case Ordering Problem* is presented and formalized; second, a method is proposed to address this problem in the context of surgical procedural training.

This paper is structured as follows: Section II presents the CVC real case used as a running example to illustrate the concepts of the paper. In Section III the problem is presented, and Section IV proposes a method to address the problem. In Section V, an alternative is provided for one of the steps of the method, using available validated medical evaluation

978-1-7281-5613-2/19/$31.00 ©2019 IEEE

preprint 10.1109/SCCC49216.2019.8966431

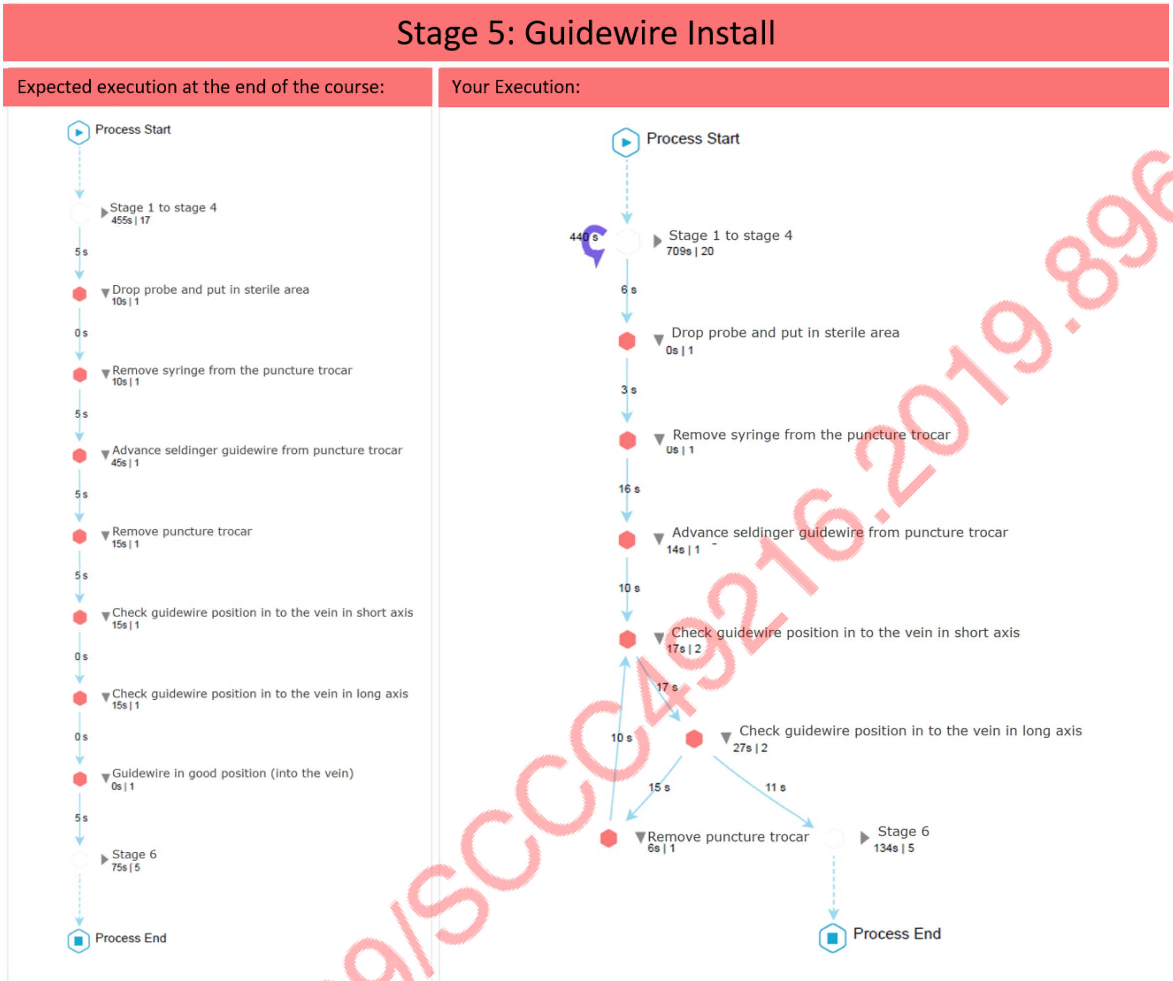

Fig. 1. Diagram of the Guidewire Install surgical procedure stage included in the feedback report from [9]. The expected execution is shown on the left side and the student's performance is shown on the right side.

preprint 10.1109/SCCC49216.2019.8966431

instruments. In Section VI a preliminary experimentation is presented to evaluate the feasibility of the proposal with both synthetic data and a real case. Finally, Section VII concludes the paper.

II. Running Example: The CVC Case

In order to illustrate the different concepts of the approach we will use the Central Venous Catheter (CVC) case as a real running example. The Central Venous Catheter case (hereinafter referred to simply as CVC) is a surgical procedure training where the students are trained about how to install a catheter using ultrasound. We released the data of a particular execution of this course as part of the Conformance Checking Challenge 2019, and it is publicly available in [13] for reproducibility purposes.

The general idea of the CVC procedure is as follows: 1) a large hollow needle (called trocar) attached to a syringe is punctured into the vein with the help of ultrasound, 2) the syringe is used to check if there is blood return (the trocar is well installed) and it is removed from the trocar if so, 3) a guidewire is inserted through the trocar, and then the trocar is removed, 4) the pathway is widen and the wire is used to advance a catheter, 5) finally, the guidewire is removed.

The procedure was modeled using other medical instruments –Checklists and Global Rating Scales [10]– designed for the same procedure [14], and it was iterated by a medical experts Delphi panel [9]. The execution data was also processed by medical experts through video observations of the procedure. Models and execution data in different formats are available in [13]. We refer the reader there for more details on the CVC case.

III. Problem Definition

The first contribution of this paper is the definition of a novel problem designed to aid in the new process-oriented medical education research line: *Optimal Case Ordering Problem*. As it was aforementioned, the final goal of the problem is to determine an order for the surgical procedural training executions (i.e., *cases*) of the students in a *course*, to aid the instructor in the analysis and evaluation of the reports derived from those executions.

In order to define the problem, we first define the necessary concepts.

Definition 3.1 (Activity, Case, and Course): Let A be the set of *activities* defined for a surgical procedure, i.e., the procedure steps to be executed. Each student execution of the procedure is represented by a *case*, i.e., a sequence of activities $c = \langle a_1, a_2, \ldots a_n \rangle$ performed by the student. A *course* of students $C = \{c_1, c_2, \ldots, c_n\}$ is represented by the set of cases, one for each student.

In a practical scenario, the student execution may include other information besides the activities and their order, e.g., the specific medical implements used in a specific activity. However, since this is a procedural perspective based approach, we can abstract from this additional information. Similarly, a case may contain other information, e.g., the name of the students or their medical discipline. In this approach we assume an identifier (e.g., k) to refer to the case c_k, identifying the student, and eventually all the additional information.

The CVC case defined 29 activities, such as *Puncture* (the trocar), *Install Guidewire*, or *Advance Catheter*. The course contained 20 students.[1] The case c_6 corresponds to the sequence of activities $\langle$*Get in sterile clothes, Prepare implements, ..., Puncture, Blood return, ..., Remove guidewire, Check flow and reflow, Check catheter position*$\rangle$. For the sake of reproducibility, we assume as case identifier the case position in the execution data provided in [13].

Given two cases, we define a distance function between them, as a representation of the procedural perspective similarity between the cases.

Definition 3.2 (Procedural Case Distance): Let c_a and c_b to be two cases of a course C, i.e., $c_a, c_b \in C$. We define the function $d(c_a, c_b) \rightarrow [0, +\infty)$ as the *distance* between two cases.

This generic distance definition can be instantiated in a wide range of different ways. In Section IV we propose a method and a function instantiation specific for the surgical procedure domain and its characteristics.

In order to find an ordering of the executions, we first define a course ordering and its total distance.

Definition 3.3 (Course Ordering and Total Distance): Let C be a course such that $C = \{c_1, c_2, \ldots, c_n\}$. A course ordering $O = \langle o_1, o_2, \ldots, o_n \rangle$ is defined as a sequence of the cases in C, i.e., $|C| = |O|$ and $\forall_{c \in C}$ exists an $o \in O$ such that $o = c$. We denote as $\mathcal{U}_O$ the universe of all possible orderings of the course.[2] Given an ordering $O = \langle o_1, o_2, \ldots, o_n \rangle$ and a distance function d, we define the *total distance* of the ordering as the sum of the distances between all two consecutive cases in the ordering, i.e., $t(O) = \sum_{i=1}^{|O|-1} d(o_i, o_{i+1})$.

The problem presented in this paper is defined as finding an order in which the total distance is minimal, i.e., the order is optimal.

Definition 3.4 (Optimal Case Ordering Problem): Let C be a course, and let d and t be the distance and total distance functions, respectively. We define the *optimal case ordering problem* as finding an ordering O for the course C such that $\forall_{O' \in \mathcal{U}_O} t(O) \leq t(O')$, i.e., there is no other case ordering with lower total distance.

Notice that the problem is different than comparing each case with the ideal model, and determining an ordering from more conformant to less conformant. The problem propose to minimize the distance between an execution and the next one, in order to aid the instructor in the report evaluation, reducing the comparison gap between two consecutive evaluations.

IV. POME Case Ordering Approach

The second contribution of this paper is the proposal of a method to instantiate the aforementioned *optimal case ordering problem* to be applied to process-oriented medical education (POME) scenarios. The method takes into account the specific characteristic of the surgical procedural training domain. First, a subset of activities (called keystone activities) are selected to represent the main procedural elements of the process. Later, the keystone activities are used to define a set of abstraction features, and a footprint of such features is extracted from each case. Next, the similarity between two cases is defined by the distance between these feature footprints. Finally, a Greedy and a Branch & Bound algorithm are proposed for finding the optimal case ordering. This section first presents each step of the method.

A. POME Keystone Selection

The first step of the approach is to select the *keystone* activities, i.e., the activities used to extract a process perspective abstraction of the cases.

Definition 4.1 (POME Keystone Selection): Let A be the set of activities of the process. We define *keystone* activities $K \subseteq A$ as the subset of activities used to abstract the process perspective of the model.

In a surgical procedure there is an heterogeneous variety of activities. For example, some activities refer to actions (e.g., puncture, suture, or install an implement), while others refer to checking activities to verify the correct state of the process (e.g., check catheter position). Although all have an important role in the process, this method proposes the use of action activities as keystone activities, as a suitable abstraction for the procedural perspective in this scenario, since the action activities are the ones that clearly advance the state of the procedure. Although this method proposes the use of action activities, in Section V we discuss alternative keystone selection criteria based in domain-specific instruments such as Checklists and Global Rating Scales [10].

In the CVC case, we select the following 7 activities as the keystone activities: *Puncture, Remove syringe, Install guidewire, Remove trocar, Widen pathway, Advance catheter, and Remove guidewire*.

[1] The course was designed with a before (PRE) and after part (POST) executions with 10 students in each part. For the sake of simplicity, in this paper we assume all 20 executions as 20 different students of the same course.

[2] Notice that to be formally correct, elements described in this paper must include their contexts, e.g., $\mathcal{U}_O^C$ to specify the course C it refers to. However, and for the sake of readability, we omitted them when the context is clear.

preprint 10.1109/SCCC49216.2019.8966431

B. POME Procedural Features Abstraction

The second step of the method is the extraction of a feature sequence for each case, representing its procedural perspective, using the keystones previously defined.

In the literature, there is a wide range of alternatives to extract features from a process execution. One of the most widely used in Process Mining is the dependency relation abstraction. This relation is used in several algorithms such as Heuristics Miner [15], and commercial tools such as Celonis [16], because it is easy to interpret for non process expert users (e.g., physycians). Given the medical nature of our scenario, we decided to use the dependency relation as the basis of our method.

Definition 4.2 (POME Procedural Features Extraction): Let K be the set of keystones. The *abstraction features* $\langle f_1, f_2, \ldots, f_n \rangle$ are defined as all possible pairs between keystones activities, i.e., $\{(k_1, k_2) | k_1, k_2 \in K\}$. We define the *feature sequence* $f(c) = \langle v_1, v_2, \ldots, v_n \rangle$ of a case c, as the sequence of integers indicating the number of occurrences each feature in the case.

In the CVC case, and given the 7 keystone activities: *Puncture (P), Remove syringe (RS), Install guidewire (GI), Remove trocar (RT), Widen pathway (WP), Advance catheter (AC), and Remove guidewire (RG)*, the number of features is $7 \times 7 = 49$, one for each pair of activities. The feature abstraction for the case c_6 corresponds to the sequence $\langle 1, 1, 1, 1, 1, 1, 0, \ldots, 0 \rangle$, where there is a 1 for the features (P,RS),(RS,GI),(GI,RT),(RT,WP),(WP,AC),(AC,RG), and a 0 for the rest.

C. POME Procedural Case Distance

The third step of the approach is to instantiate the function to define the procedural distance between two cases (cf. Definition 3.2), i.e., define the degree of similarity between the feature abstraction sequence of two cases. In the literature, there is a wide range of well known distance metrics that can be applied to measure the similarity between sequences/vectors of integers, e.g., Euclidean distance, Hamming distance, Levenshtein distance, among others [17]. This approach proposes the use of the general purpose distance –Manhattan distance [18]– given that is easily applied to all surgical procedures.

D. POME Optimal Case Ordering Computation

The last step of the approach is to compute an optimal ordering, i.e., solve the Optimal Case Ordering Problem (cf. Definition 3.4). It is easy to see that this is a computationally complex problem, with high similarity with the *Travel Salesman Problem (TSP)*: in this problem, each city is a case, the distance between cities is the distance between cases, and the tour visiting every city is the ordering of each case.

Strategies designed to solve the TSP could be easily translated to the case ordering problem. This approach proposes first the use of a greedy strategy that provides a (possible) good solution almost instantaneously. The greedy strategy proposed is based on the *Nearest Neighbour (NN)* heuristic: given a case, the greedy strategy chooses the closest case to add to the partial solution. In the TSP problem, this algorithm quickly yields an effectively short route, e.g., for N cities randomly distributed on a plane, the algorithm on average yields a path 25% longer than the shortest possible path [19]. In our case, the best effort greedy algorithm achieves similar results, even nearer in most cases. For example, in the CVC case, the proposed greedy algorithm finds a 41 distance solution, being 39 the optimal distance.

Notice that, although optimization problems such as those are not feasible for non-small number of cities/cases, the method proposed in this paper was designed for the POME surgical procedures training scenario, i.e., for logistic and pedagogical reasons in such domain courses are typically of 6-12 students. Moreover, it is not a realistic assumption to consider that a single instructor will review a higher number of procedural reports in one sitting. Additionally, depending on the procedural abstraction used, several students may have the same feature sequences (e.g., specially the correct order of the procedure commonly performed by the most advanced students), being able to aggregate these cases, reducing the computation time. Thus, this approach proposes an additional algorithm to compute the exact optimal ordering. The proposed method is based on the Branch & Bound algorithms [20]: first, an upper-bound is computed using a fast greedy algorithm (in our case, the NN greedy described above), then the search space is explored but pruning those paths with guaranteed worst solutions. This bound is constantly updated during the exploration [20].

This algorithm is preferred when immediately response is not required, since it guarantees the optimal ordering. For example, in the CVC case, the computation time using this algorithm was around 2 minutes. Since this is only done once before the instructor evaluations, this is a reasonable alternative.

V. Instrument-based Guidelines for Keystone Selection

In the previous section, a keystone activity selection was proposed, where the semantic of the activities is used, i.e., action activities are selected due to their goal of advance the procedure, where other activities such as checks and preparations are discarded. Although this approach provides meaningful results, alternative keystone selection methods could be proposed. In particular, this section discusses keystone selection based on available evaluation instruments.

It is easy to see that, since the process perspective is a novel and complementary analysis technique, current surgical procedure training uses well-established and validated instruments to evaluate the students. The most commonly used are Checklists (CL) – where a medical expert indicates if a step has been performed or not – and Global Rating Scales (GRS) – where each element is evaluate in a scale which levels depend on the item evaluated [10].

The proposed approach matches items from the CL and GRS, and activities in the process. Process activities with a corresponding match are selected as keystones (because

preprint 10.1109/SCCC49216.2019.8966431

Area is cleaned with chlorhexadine
Resident gets in sterile gown, gloves, hat and mask
Area is draped in usual sterile fashion (must be full body drape)
The vein is localized
The skin is anesthetized with 1% lidocaine in a small wheal
The deeper structures are anesthetized
Using the large needle, cannulate the vein while aspirating
Remove the syringe from the needle
Advance the guidewire into the vein no more than approximately 12–15 cm
Knick the skin with the scalpel to advance the dilator
Advance the dilator over the guidewire and dilate the vein
Advance the triple lumen over the guidewire
Never let go of the guidewire
Once the catheter is inserted remove the guidewire in its entirety
Advance the catheter to approx 14–16 cm on the right side
Ensure there is blood ow/ush each port
Secure the catheter in place (suture or staple)
Maintain sterile technique
Ultrasound set up

TABLE I
CHECKLIST USED BY THE INSTRUCTOR TO ASSESS THE STUDENTS PERFORMANCE.

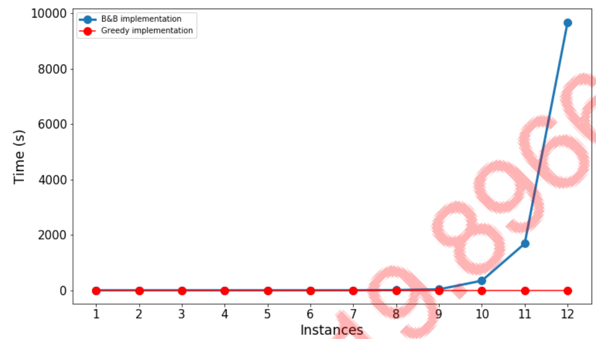

Fig. 2. Performance comparison by course cases (instances) between Greedy and Branch&Bound algorithms.

current medical evaluation focuses on such steps), while not matched activities are discarded. The matches are done using the semantics of the items and activities. Three types of matches are possible: 1-to-1 matches (where a CL/GRS item is matched to exactly one process activity), 1-to-n matches (where a CL/GRS item refers to more than one activity), and n-to-1 matches (where several items analyze the execution of the same activity). In the CVC case, a Checklist was available (see Table I for the specific CVC item list [14]). Item *Resident gets in sterile gown, gloves, hat and mask,* can be matched 1-to-1 with the process activity *Get in sterile clothes*. On the other hand, *Advance the guidewire into the vein no more than approximately 12–15 cm, Advance the triple lumen over the guidewire* and *Never let go of the guidewire*, are matched n-to-1 with the activity *Guidewire install*. Finally, *Anatomic identification, Doppler identification* and *Compression identification* process activities are matched 1-to-n with the Checklist item *The vein is localized*, and model could be simplified with a *Vein Identification* activity. The resulting keystones for the CVC are *Prepare implements, Get in sterile clothes, Drap puncture area, Ultrasound configuration, Put sterile gel, Position probe, Vein Identification, Anesthetize, Puncture, Blood return, Remove syringe, Guidewire install, Widen pathway, Advance catheter, Remove guidewire, Check flow and reflow* and *Check catheter position*.

The proposed guidelines could be refined, introducing the notion of weights for each match, and defining a threshold to establish whether an activity is selected as a keystone. For instance, activities with n-to-1 matches has more weight (assuming that activities that are evaluate from different perspectives has more interest for the instructors), or matches from GRS have higher weight (since the instructors are not only interested if the activity is performed but the level of proficiency on its performance).

VI. PRELIMINARY EXPERIMENTAL EVALUATION

In this section, we present some preliminary experimental evaluation of the approach, as it is presented in Section IV. The evaluation includes experiments with synthetic data and a real case validated by a medical expert. Notice that, giving the workshop nature of the venue, this section provides only a preliminary evaluation, providing only intuitions of the validity of the approach. A more complete evaluation is required, including experiments with several experts, but for space and time constrains this is left as a future work.

In the first experiment, we illustrate the effects in time accuracy of the approach, regarding the number of cases in a course. For that, incremental synthetic CVC-inspired cases were randomly generated, and the two algorithms were tested: Greedy, and Branch&Bound (B&B). For each algorithm we provide the computation time, and the resulting total distance of the ordering found. Experiments were executed on an Intel Core i5 processor running at 2GHz and 8GB of RAM. The approach was coded in Python.

Performance results are presented in the Figure 2. As expected, the B&B presents a exponential behavior, showing a significant time increase after 9 cases. However, 9 cases for a surgical course is a realistic number, specially since B&B provides an optimal ordering. On the other hand, as expected, the Greedy algorithm presents a linear time, providing results almost instantaneously. That is also the case for larger courses (15,20,25,...), although the figure only shows until 12 for comparison reasons with the B&B. Although the Greedy does not guarantee an optimal, in most cases the provided solution is optimal or close to, as Figure 3 shows. The figure compares the B&B total distance (the optimal) with the Greedy best effort solution, for all the synthetic cases defined. In half of the cases the solution reported is optimal, and in the others the difference only ranges between 1 and 5. As recommendation, Greedy should be used with larger courses or whenever instantaneous answer is required (e.g., when different distance criteria want to be compared), while B&B should be used for average size courses and when it is necessary to be performed only once before the instructor evaluations.

preprint 10.1109/ISCC49216.2019.8966431

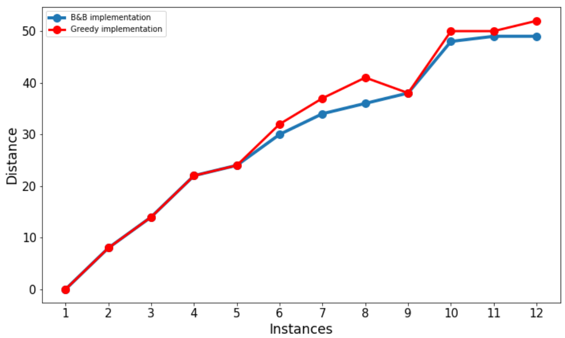

Fig. 3. Total distance comparison by course cases (instances) between Greedy and Branch&Bound algorithms.

In the second experiment, a real case was used: the CVC case presented as a running example. The course consisted of 20 students, and the keystones and functions were used as described in Section IV. Feature abstractions were extracted from the 20 students. We detected that 11 out of 20 had the same feature sequence, mostly the more advance students who performed the procedure as expected. Those cases were aggregated in 1, resulting in 10 executions in total. The Greedy algorithm took *0.2 seconds*, and reported a solution of *distance 41*. On the other hand, the Branch & Bound took *2 minutes and 13 seconds*, reporting an optimal solution of *distance 39*. As expected, the time difference is considerable, although both provide results in a reasonable time, while the best-effort solution is only 2 points below the optimal.

Table II shows the resulting ordering for each method. Cases are identified by their real CaseID, and an additional StudentID for the sake of readability of this table. Additionally to the orderings provided by the algorithm, a medical expert and instructor of the CVC course was asked to provide an ordering based on the case similarity, also reported in Table II.

Table II shows similarities between the orderings obtained by both algorithms, and some minimal differences. Greedy places S_6 at the beginning, before the sequence $S_1, \ldots, S_5$, and swaps the order of the students S_9 and S_8. In order to illustrate the differences between the keystone activities among consecutive students in the obtained orderings, Figure 4 shows a diagram for the execution of the keystone activities for each of the 5 students of the sequence $S_1, \ldots, S_5$, mentioned before. It can be observed that the first case has a greater distance, $d = 8$, to the following case compared to the distances between the other cases, which is reflected in the fact the first diagram is quite different to the other ones; therefore, the first case could be considered as an outlier. In turn, without considering this outlier case, a pattern is observed in the ordering of the remaining cases, since they differ in the amount of keystone activities that are executed before a rework; the rework occurs earlier while descending in the diagrams.

StudentID / CaseID	B&B	Greedy	Expert
S_1 = 1539314415211video_2.1_CVC	S_1	S_6	$\cancel{S_1}$
S_2 = 1539316889981video_3.1_CVC	S_2	S_1	S_6
S_3 = 1539832275246video_4.8_CVC	S_3	S_2	S_5
S_4 = 1547698148253video_1.h_CVC	S_4	S_3	S_4
S_5 = 1539737717686video_3.3_CVC	S_5	S_4	$\cancel{S_7}$
S_6 = 1539739275781video_4.5_CVC	S_6	S_5	S_3
S_7 = 1547683734202video_1.c_CVC	S_7	S_7	S_2
S_8 = 1539302414925video_1.3_CVC	S_8	S_9	S_9
S_9 = 1539734942389video_3.2_CVC	S_9	S_8	S_8
S_{10} = 1548037113729video_2.d_CVC	S_{10}	S_{10}	S_{10}

TABLE II
ORDERING OF THE REAL CVC CASE REPORTED BY BRANCH & BOUND, GREEDY, AND A MEDICAL EXPERT.

Regarding the ordering proposed by the medical expert, it is important to keep in mind that the medical expert considered S_1 and S_7 as outlier cases, since the former does not correspond to a logical sequence of keystone activities (first case in Figure 4), and the latter is the only case in which the student does not execute all the activities required to carry out the procedure. When asking the medical expert about the criteria used for establishing the ordering, the medical expert highlights the identification of outliers (the two described previously), and grouping cases taking into account semantic aspects. An example of grouping is the set of cases $S_2, \ldots, S_5$ shown in Figure 4, without considering the outlier S_1. The same ordering appears in the ordering of the medical expert, but in the reverse order $S_5, \ldots, S_2$ (excluding the outlier S_7).

VII. Conclusions and Future Work

This paper addressed the challenge of providing a favorable student executions ordering, to maximize the similarity between consecutive executions, so instructors reviewing them could benefit from the comparative aid of the previous analysis. First, the paper defined and formalized the problem, and then, it proposed a method to compute the ordering tailored to the surgical procedure training scenario.

As future work, we would like to enhance our approach considering the two criteria proposed by the medical expert: allowing to detect outliers, and allowing to group cases before ordering each group independently. Moreover, we plan to conduct a more complete experimental study involving several medical experts.

Acknowledgment

This work is partially supported by CONICYT FONDECYT 11170092, CONICYT REDI 170136, VRI-UC INTERDISCIPLINA II170003, and FOND-DCC 2017-0001.

References

[1] E. Rojas, J. Munoz-Gama, M. Sepúlveda, and D. Capurro, “Process mining in healthcare: A literature review,” *Journal of Biomedical Informatics*, vol. 61, pp. 224–236, 2016.
[2] W. M. P. van der Aalst, *Process Mining - Data Science in Action*. Springer, 2016.
[3] R. Mans, H. A. Reijers, M. van Genuchten, and D. Wismeijer, “Mining processes in dentistry,” in *ACM IHI 2012*. ACM, 2012, pp. 379–388.
[4] R. C. Basole and et.al, “Understanding variations in pediatric asthma care processes in the emergency department using visual analytics,” *JAMIA*, vol. 22, no. 2, pp. 318–323, 2015.

preprint 10.1109/SCCC49216.2019.8966431

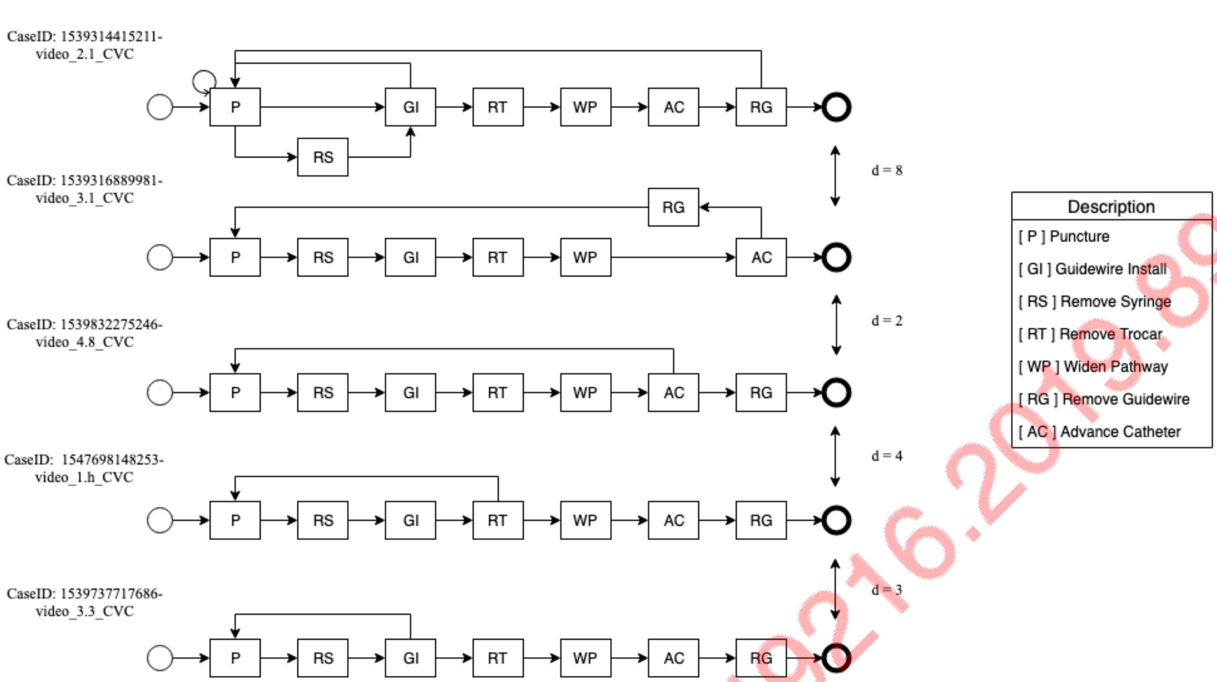

Fig. 4. Executions of keystone activities in real case analysis

[5] F. Caron, J. Vanthienen, K. Vanhaecht, E. van Limbergen, J. D. Weerdt, and B. Baesens, "Monitoring care processes in the gynecologic oncology department," *Comp. in Bio. and Med.*, vol. 44, pp. 88–96, 2014.

[6] D. Neumuth, F. Loebe, H. Herre, and T. Neumuth, "Modeling surgical processes," *Artificial Intelligence in Medicine*, vol. 51, no. 3, pp. 147–161, 2011.

[7] T. Neumuth, P. Jannin, G. Strauß, J. Meixensberger, and O. Burgert, "Validation of knowledge acquisition for surgical process models," *JAMIA*, vol. 16, no. 1, pp. 72–80, 2009.

[8] R. Lira and et al., "Tailored process feedback through process mining for surgical procedures in medical training: The central venous catheter case," in *BPM 2018*, ser. LNBIP, vol. 342. Springer, 2018, pp. 163–174.

[9] ——, "Process-Oriented Feedback through Process Mining for Surgical Procedures in Medical Training: The Ultrasound-Guided Central Venous Catheter Placement Case," *Int. J. Environ. Res. Public Health*, vol. 16(11), no. 1877, pp. 1–23, 2019.

[10] J. S. Ilgen, I. W. Ma, R. Hatala, and D. A. Cook, "A systematic review of validity evidence for checklists versus global rating scales in simulation-based assessment," *Medical education*, vol. 49, no. 2, pp. 161–173, 2015.

[11] A. K. Jain, R. C. Dubes *et al.*, *Algorithms for clustering data*. Prentice hall Englewood Cliffs, NJ, 1988, vol. 6.

[18] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.

[12] D. A. Schön, "Educating the reflective practitioner," 1987.

[13] J. Munoz-Gama, R. de la Fuente, M. Sepúlveda, and R. Fuentes, "Conformance Checking Challenge 2019," 4TU.Centre for Research Data https://doi.org/10.4121/uuid:c923af09-ce93-44c3-ace0-c5508cf103ad.

[14] M. Corvetto, J. Pedemonte, D. Varas, C. Fuentes, and F. Altermatt, "Simulation-based training program with deliberate practice for ultrasound-guided jugular central venous catheter placement," *Acta Anaesthesiologica Scandinavica*, vol. 61, no. 9, pp. 1184–1191, 2017.

[15] A. J. M. M. Weijters and J. T. S. Ribeiro, "Flexible heuristics miner (FHM)," in *CIDM 2011*. IEEE, 2011, pp. 310–317.

[16] J. Geyer-Klingeberg, J. Nakladal, F. Baldauf, and F. Veit, "Process mining and robotic process automation: A perfect match," in *BPM 2018*, ser. CEUR Workshop Proceedings, vol. 2196. CEUR-WS.org, 2018, pp. 124–131.

[17] I. Bonet, A. Rodríguez, R. G. Ábalo, M. M. García, Y. Saeys, and A. Nowé, "Comparing distance measures with visual methods," in *MICAI 2008*, ser. LNCS, vol. 5317. Springer, 2008, pp. 90–99.

[19] D. S. Johnson and L. A. McGeoch, "The traveling salesman problem: A case study in local optimization," *Local search in combinatorial optimization*, vol. 1, no. 1, pp. 215–310, 1997.

[20] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.

preprint 10.1109/SCCC49216.2019.8966431