Inf Syst Front (2017) 19:1101–1121
DOI 10.1007/s10796-016-9715-1

Design it like Darwin - A value-based application of evolutionary algorithms for proper and unambiguous business process redesign

Patrick Afflerbach[1] · Martin Hohendorf[2] · Jonas Manderscheid[2]

Published online: 8 November 2016

Abstract Business process management (BPM) is an acknowledged source of corporate performance. Despite the mature body of knowledge, computational support is considered as a highly relevant research gap for redesigning business processes. Therefore, this paper applies Evolutionary Algorithms (EAs) that, on a conceptual level, mimic the BPM lifecycle – the most popular BPM approach – by incrementally improving the status quo and bridging the trade-off between maintaining well-performing design structures and continuously evolving new designs. Beginning with describing process elements and their characteristics in matrices to aggregate process information, the EA then processes this information and combines the elements to new designs. These designs are then assessed by a function from value-based management. This economic paradigm reduces designs to their value contributions and facilitates an objective prioritization. Altogether, our triad of management science, BPM and information systems research results in a promising tool for process redesign and avoids subjective vagueness inherent to current redesign projects.

Keywords Evolutionary algorithms · Genetic programming · Business process redesign · Business process management · Computational process management · Value-based business process management

✉ Jonas Manderscheid
jonas.manderscheid@fim-rc.de

Patrick Afflerbach
patrick.afflerbach@fim-rc.de

Martin Hohendorf
MHohendorf@gmx.de

[1] Project Group Business & Information Systems Engineering of the Fraunhofer FIT, Universitaetsstrasse 12, 86159 Augsburg, Germany

[2] Research Center Finance & Information Management, University of Augsburg, Universitaetsstrasse 12, 86159 Augsburg, Germany

1 Introduction

Process orientation is an accepted paradigm of organizational design with a proven impact on corporate performance (Kohlbacher and Reijers 2013). An essential management task that organizations have to continuously execute when subscribing themselves to this proven paradigm is process redesign. It aims at increasing effectiveness and efficiency of processes by adapting the actual process design to changes in the organizational environment. Thereby, the interpretation of the term process design varies with respect to the level of abstraction. It ranges from a very high-level interpretation as an operational sequence description of executed activities and their chronological order to a very detailed interpretation as a process model which considers every possibility that may affect the way of how work is performed. This paper follows an in-between interpretation of a *process blue-print* and define a process design as a description of activities, their chronological and their logical order. As process redesign is often considered as the most value-creating activity within BPM (Dumas et al. 2013; Zellner 2011), extensions of the scientific and practical tool-kit for such redesigns are still in high demand (van der Aalst 2013). Although the constant attention from industry and academia resulted in a plethora of mature approaches, methods, and tools (Harmon and Wolf 2014; van der Aalst 2013; Vanwersch et al. 2016), most redesign approaches are of qualitative nature and heavily rely on human intuition as their source of innovation (Hofacker and Vetschera 2001). Brainstorming sessions and iterative discussions are the pillars of the so-called *creative redesign approach* (Limam Mansar et al. 2009), although it is known that such discussions may bias choices and neglect alternatives. As a consequence, practical decision-makers are in deep need of computational support for the redesign act to overcome the inherent subjective vagueness (Sharp and McDermott 2008; Zellner 2011). From a scientific perspective, many scholars confirm the relevance of this research topic and denote the lack of computational redesign support as an important and current research gap (van der Aalst 2013; Vergidis et al. 2008; Zellner 2011).

Considering the success of computational intelligence (CI) in design and optimization problems from the business world, this paradigm seems promising. The abilities to cope with complex processes and a mass of data (gathered by workflow management or business intelligence systems, see van der Aalst (2013)) as well as to reduce uncertainty and subjective vagueness underlines the attractiveness of the paradigm. Further, applications of evolutionary algorithms (EA) as a prominent representative of the CI-tool-kit have already shown their potential in solving BPM problems. For example, Low et al. (2014), Richter-Von Hagen et al. (2005), and Zhou and Chen (2003) use EAs to assign resources to process activities. Vergidis et al. (2012) even utilizes the power of EAs and CI to improve process designs. However, current works do not unfold the complete potential of EAs: Their multi-objective perspectives lead to ambiguous solutions. Performance issues restrict the complexity of the process under investigation. Essential characteristics as decision nodes and the corresponding conditions are out of scope. This is why this paper investigates the following research question: *How can organizations leverage CI to redesign their processes while accounting for the essential process elements?*

In order to address this research question, this paper develops an EA-application in a broader sense and translates the real-world problem of BPM to the computational world (and back again) for solving it by CI. This allows a dynamic design of processes and supports practitioners in validating and evaluating design alternatives. Our application considers the essential process elements (e.g., activities, objects and their logic connectivity), which is the key challenge in the translating part. As research method, design science research (DSR) paradigm is chosen as EAs fulfill the criteria of a valid DSR artefact type (March and Smith 1995). As justificatory knowledge, this study draws from a theoretical triad of CI as representative of IS research, value-based management (VBM) from management sciences and BPM as an intersecting discipline. BPM and CI provide the theoretical foundation for our application. As the evolutionary design of processes is a proven best practice in BPM, the transfer of the evolutionary way to the computational level has a sound theoretical foundation (Dumas et al. 2013). As an acknowledged theory for corporate and process decision-making, VBM serves for evaluating the computed redesign alternatives (Buhl et al. 2011; vom Brocke and Sonnenberg 2015).

Following the DSR methodology as per Peffers et al. (2007), this study discusses the identification of and motivation for the research problem, objectives of a solution, design and development, and evaluation. Section 2 outlines the development of computational intelligence in BPM to position the contribution of our work. Section 3 derives design objectives from the business requirements (*objectives of a solution*) and provides relevant justificatory knowledge. Section 4 outlines the research idea and evaluation strategy. Section 5 introduces the design specification of the EA application (*design and development*). Section 6 reports on our evaluation activities (*evaluation*). The authors conclude in section 7 by pointing to limitations and future research possibilities.

2 Computational intelligence in the history of BPM

BPM is an integrated system for handling organizational performance, regulatory compliance, and service quality by managing processes (Dumas et al. 2013; Hammer 2015). In other words, it is "the art and science of overseeing how work is performed […] to ensure consistent outcomes and to take advantage of improvement opportunities" (Dumas et al. 2013, p. 1). Thereby, it combines knowledge from computer and management sciences (van der Aalst 2013). Following the historic overview on the evolution of BPM by van der Aalst (2013), the role of CI goes back to the process improvement postulation in the mid-nineties (Hammer and Champy 1993) when BPM finally found its way into information systems (IS) research. Workflow management systems (WFMS) became available and computational BPM primarily focused on automation with little support regarding the analysis, flexibility, and management of processes. Today, the scientific lens of BPM increasingly shifts from an operational to an analytical orientation. With a broader scientific horizon, it now includes controlled process execution and process redesign.

Concerning the mission of providing practical support on redesign projects, the BPM community has produced a variety of tools that can act as facilitators or enablers for the identification and implementation of improved process designs. However, there is still little support for computer-based and automatic generation of innovative design ideas (Bernstein et al. 2003). Scholars mainly provide qualitative techniques such as brainstorming (Kettinger et al. 1997). Although also more advanced techniques such as *RePro* begin to evolve (Vanwersch et al. 2015), only few works respond to the need of computational support. To list some examples: Case-based reasoning (CBR) is a first approach leveraging computational abilities to create new process designs by searching analogies to successful redesign projects implemented in the past (Min et al. 1996). The *process recombinator tool* by Bernstein et al. (2003) proposes new process designs based on a list of core activities. Although providing computational support for the construction and identification of new, promising designs, this tool is only semi-automatic as the selection of the most satisfactory process design is delegated to the user. The *KOPer tool* by Nissen (1998) identifies problematic process structures or fragmented process flows to find designs dealing with these so called process pathologies. However, the prioritization and realization of redesigns also remains a manual task. Limam Mansar et al. (2009) build on CBR and the KOPer tool. They derive *best practices for process redesign* and provide empirical evidence to process managers. Besides, some applications of EAs for process redesign have emerged, presenting EAs as a promising approach to fill the gap of automated support: Zhou and Chen (2003) and Richter-Von Hagen et al. (2005) optimize resource assignments with regard to multiple performance objectives, whereas Richter-Von Hagen et al. (2005) have a distinct focus on knowledge-intensive processes. Vergidis et al. (2012) evaluate alternative process designs varying in size and activities due to their expected performance in fulfilling multiple objectives and resulting in a set of not-dominated designs. Low et al. (2014) use EAs to redefine starting times of activities and reallocate resources from a cost-based view. Although process performance is often considered from various perspectives like time, quality and costs, the integration of this multiplicity into EAs often comes along with performance and complexity restrictions.

Briefly, practitioners and academia have recognized the importance of tool-support for process redesign and provide first approaches that use artificial intelligence algorithms. Nevertheless, we can justify applications of CI in BPM as meaningful research problem: Although IT and computational intelligence already find applications in the design of new process alternatives and help to make the design process more easily, more cost-effectively, quicker, more systematically, and more robust against subjective vagueness, the technical task of generating new process designs is still in its infancy (Limam Mansar et al. 2009). This paper addresses this research gap by enhancing existing approaches by implementing additional process elements, establishing unambiguous redesign objectives to deal with the increasing complexity of today's processes.

As the further development of the existing approaches intends to design and implement a new and innovative artefact (e.g., models, methods, constructs, instantiations, and design theories or in our case computational intelligence tools for process redesign) (Hevner et al. 2004; March and Storey 2008), it could follow the design science research (DSR) paradigm as theoretical fundament (March and Storey 2008). The DSR methodology as per Peffers et al. (2007) proceeds in six steps: *identification of and motivation for the research problem*, *definition of the objectives of a solution*, *design and development*, *demonstration*, *evaluation*, and *communication*. As we already identified and motivated a meaningful DSR problem, we proceed with step 2.

3 Design objectives and justificatory knowledge

In order to accomplish the second step of DSR (Peffers et al. 2007), we need to derive design objectives ((O.1) – (O.3)) from justificatory knowledge. In general, design objectives help to assess whether an artefact properly solves the identified research problem. As justificatory knowledge, we refer to BPM and to VBM. As processes and their elements are the essentials of process redesign, we define the first design objective:

(O.1) Process Elements: To redesign processes, it is necessary to consider the key elements of processes: activities, connections and routing decisions.

A process is defined as a collection of inter-related events, activities, and decision points that involve a number of actors (or resources) and objects, and that collectively lead to an outcome (Dumas et al. 2013). The specific order of activities describes how the involved actors perform their activities across time and place (Davenport 1993). Their executions may be sequential or happen in parallel. The used objects can be tangible (e.g., precious metals) or intangible (e.g., customer data) goods. They serve as in- and/or output in their original or modified forms. Each set of activities in a specific order represents a process path. In the case of necessary distinctions, defined conditions decide on the right path (routing decision). Summarizing, with design objective O.1, the combinations of activities, objects, conditions, etc. form realistic process designs with different levels of complexity.

As the implemented design in turn influences the overall process performance, the design candidates provide a basis for prioritization (Limam Mansar et al. 2009). To provide concrete guidance for redesign initiatives and to support a clear corporate decision-making, we define the second design objective:

(O.2) Value-Based Management: To prioritize process redesign, it is necessary to cater for cash flow effects and the time value of money. Moreover, the involved decision-makers' risk attitude must be considered.

In the context of BPM, organizations normally use performance indicators together with desired target values (benchmarks) and admissible value ranges (Leyer et al. 2015) to assess the performance of a process. Process performance indicators can be grouped via the Devil's Quadrangle, a framework comprising a time, cost, quality, and flexibility dimension (Reijers and Limam Mansar 2005). The Devil's Quadrangle is so-named because improving one dimension weakens at least one other, disclosing the trade-offs to be resolved during redesign to prevent from ambiguous design prioritizations each fulfilling another objective.

To resolve the partly conflicting nature of these performance dimensions via integrated performance indicators, process decision-making at least devoted increasing attention to value-based management (vom Brocke and Sonnenberg 2015). It is the guiding paradigm of corporate decision-making in economic research and practice (Buhl et al. 2011). VBM strives for a sustainably evolution of the firm value from a long-term perspective (Ittner and Larcker 2001; Koller et al. 2015). Thereby, it extends the shareholder value approach that was established by Rappaport (1986) and elaborated by Copeland et al. (1994) as well as by Stewart and Stern (1991). Its long-term perspective makes VBM compliant with the more general stakeholder value approach (Danielson et al. 2008). For VBM to be completely established, all corporate activities and decisions must be orientated at maximizing the firm value. As key requirements – and consequently as design objective O.2, organizations must quantify the firm value on the aggregate level and the value contributions of individual assets and decisions by regarding their cash flow effects, the time value of money, and the decision-makers' risk attitude (Buhl et al. 2011). The valuation functions that are typically used for this quantification purpose originate from investment and decision theory and consider the decision situation and the decision-makers' risk attitude (Buhl et al. 2011; Damodaran 2012).

The most prominent methods in BPM that leverage the essentials of VBM for solving BPM problems are goal-oriented BPM (Neiger and Churilov 2004a), value-focused BPM (Neiger and Churilov 2004b; Rotaru et al. 2011), value-driven BPM (Franz et al. 2011), value-oriented BPM (vom Brocke et al. 2010), and value-based BPM (Bolsinger 2015). Particularly, process-related decisions based on value-oriented or value-based BPM solve the problem of the partly conflicting, multi-objective nature of performance dimensions by compiling it into the integrated, single-objective measure of a process' value contribution (Buhl et al. 2011). Both methods also consider cash flows and the time value of money. Whereas, value-oriented BPM has a stronger focus on the financial perspective and the pure cash flows in terms of the payment structure (Bolsinger 2015), value-based BPM uses the valuation functions as analytical lenses to compare process alternatives (Bolsinger 2015). In line with our intention to prioritize design alternatives, value-based BPM is qualified as guiding paradigm. Not least, ever more approaches adopt value-based BPM to support process design in an economically well-founded manner while comparing design alternatives and/or proposing improvement recommendations (Bolsinger 2015; Bolsinger et al. 2015; vom Brocke et al. 2010). Further approaches integrate the financial and non-financial performance effects that capture how work is organized and structured within the central measure of process cash flows (Afflerbach et al. 2014; Linhart et al. 2015a; Linhart et al. 2015b). As the value contribution of processes depend on the tasks and paths included in process models as well as on the tasks' monetized performance effects, methods such as that proposed by Bolsinger (2015) help aggregate multi-dimensional task and path characteristics to cash flows.

As the overall process performance varies over time owing to the constantly changing environment and, consequently, the implemented process design has to keep pace, we define the third design objective:

(O.3) Evolutionary Redesign: Computational redesign should follow an evolutionary logic to be in line with known best practices and to reduce organizational resistance.

Process redesign as the most important and valuable phase of the BPM lifecycle (Zellner 2011) evolved as an everyday task (Doomun and Vunka Jungum 2008). In regards to these redesign initiatives, companies face a technical and a socio-cultural challenge (Reijers and Limam Mansar 2005). The technical challenge relates to the identification of new process design or structures. Despite the methodological plethora for process redesign, there is still less guidance and support by means of techniques and best practices (Reijers and Limam Mansar 2005; Sharp and McDermott 2008; Valiris and Glykas 1999). The few existing approaches and the conditions to be met are too complex (Limam Mansar et al. 2009). Therefore, the tools fail to support redesign (Nissen 2000). The socio-cultural challenge originates from the organizational effects on the involved people. Many redesign initiatives struggle with organizational resistance while incorporating the newly designed processes into working practice (Wastell et al. 1994). However, only the intended use that is aligned to the strategic and operational goals of the firm may realize the value of redesign (Agarwal and Karahanna 2000). Otherwise it is worthless. To foster acceptance among practical decision-makers, it is crucial that the computational support follows a comprehensible logic in deriving new process designs. Design objective O.3 addresses both perspectives of the socio-economic challenge in a dynamic environment. As most accepted approach for process redesign (Dumas et al. 2013), the BPM lifecycle and its evolutionary, incremental procedures represent a suitable foundation for the DSR artefact.

4 Research idea and evaluation strategy

In the design and development phase of our DSR project (cf. Peffers et al. 2007), we combine ideas from IS research, management science and BPM (as the intersecting discipline) to develop an application constructed for identifying promising redesign alternatives. BPM captures the essentials of the research problem in terms of modelling the object for optimization. CI in general and EAs in particular assist with creating new designs of the process under investigation to provide a pool of design alternatives for a deliberate choice (Keeney and Raiffa 2003). VBM complements our application by providing a suitable valuation function to prioritize the pre-constructed alternatives from the EA. The fundamentals for the integration of these diverse research directions are our programming logic for transforming processes as real-world objects into artificial, algorithmic objects and our EA customization towards the requirements from the business side. This triad of research disciplines is necessary as the redesign problem per se is that complex that each discipline separately cannot meet the underlying complexity.

When developing such a business application of CI, we adhere to the following blue-print: We first choose the appropriate algorithm from the broad tool-kit of CI based on theoretical reasoning (sections 5.1). We then proceed with the problem representation and bring processes to the computational level based on LISt programming (LISP) and attribute matrices (sections 5.2). This is the key challenge of our application as it requires the orchestration and synthesis of the three research disciplines. Finally, we customize the EA in its core functions: the creation of the initial population, the evaluation of individual organisms, the selection and reproduction mechanisms (section 5.3). Within this section, we operationalize an acknowledged valuation function used in VBM as fitness function. Complying with the requirements of VBM, the fitness function considers the cash flow and risk effects of a redesign candidate, the time value of money, and the involved decision-makers' risk attitude.

To demonstrate and evaluate our artefact, we follow Sonnenberg and vom Brocke's (2012) framework of evaluation activities in DSR. This framework considers ex-ante/ex-post and artificial/naturalistic evaluation (Pries-Heje et al. 2008; Venable et al. 2012). Ex-ante evaluation is conducted in advance, ex-post evaluation after the instantiation of the algorithm, e.g., by means of a prototypical implementation. Naturalistic evaluation demands the judgement of the artefacts in real life. To validate our design specifications, we apply an ex-post evaluation (EVAL3) that assess the usefulness of the artefact instantiations. We implemented the artefact in Microsoft Excel (MS Excel) and Visual Basic for Applications (VBA).

5 Computational process redesign

We use the concepts of CI to design an algorithm supporting the process redesign problem. To match CI capabilities as problem solution and BPM requirements as problem domain, we are confronted with decisions about the appropriate algorithm, about design elements and about constructional aspects of the chosen CI algorithm (Koza 1992). Design decisions cover requirements from the problem domain, its representation and objects for optimization, as well as the representation of the design solutions. Constructional aspects relate to the population concept and the evaluation of solutions.

5.1 Evaluation of an appropriate CI approach for process redesign

CI provides a set of nature-inspired computational methodologies and approaches close to the human way of reasoning (Rutkowski 2008; Siddique and Adeli 2013). To find an appropriate support for process redesign, it is necessary to understand the evolutionary nature of processes and their management. As an intermediate step, we draw parallels to the biological evolution (Darwin 1859; Mendel 1866) as basis for the identification of a nature-inspired problem solution that follows an evolutionary logic to be in line with the well-known BPM lifecycle as problem domain and to reduce organizational resistance (see design objective O.3).

The aim of the BPM lifecycle, which is the most prominent redesign approach in practice, is analogous to the reproduction cycle in nature: an improved generation of objects. Whereas these objects are organisms (e.g., human individuals) in nature, BPM operates on processes. Their appearance are their process models and their organs are connections, activities, and objects. The phases of the BPM lifecycle (Dumas et al. 2013), i.e., identification, discovery, analysis, redesign, and implementation as well as monitoring and controlling, correspond to the phases of the evolutionary reproduction, i.e., offspring, natural selection, sexual selection and reproduction as shown in the inner part of Fig. 1. We explain the parallels between the two concepts below:

(1) Both cycles start with an object that represents a solution according to the respective objectives – viable organisms or well-performing process designs where their performances determine survivability. If distinctive characteristics give an object an edge over competitors, it is more likely to propagate in following generations (Darwin 1859). While in nature, an organism has to compete with others about scarce natural resources for survival, process designs compete in terms of effectiveness and efficiency.

(2) Every object is constantly evaluated according to its goal fulfillment. Vitality and fertility of sexual partners in nature (Darwin 1859) versus performance behavior in BPM.

(3) Reproduction (or redesign in BPM) combines or replaces the best objects and modifies them via recombination and mutation (Darwin 1859; Mendel 1866). While recombination combines the genetic material of selected objects, mutation carries out random changes to create

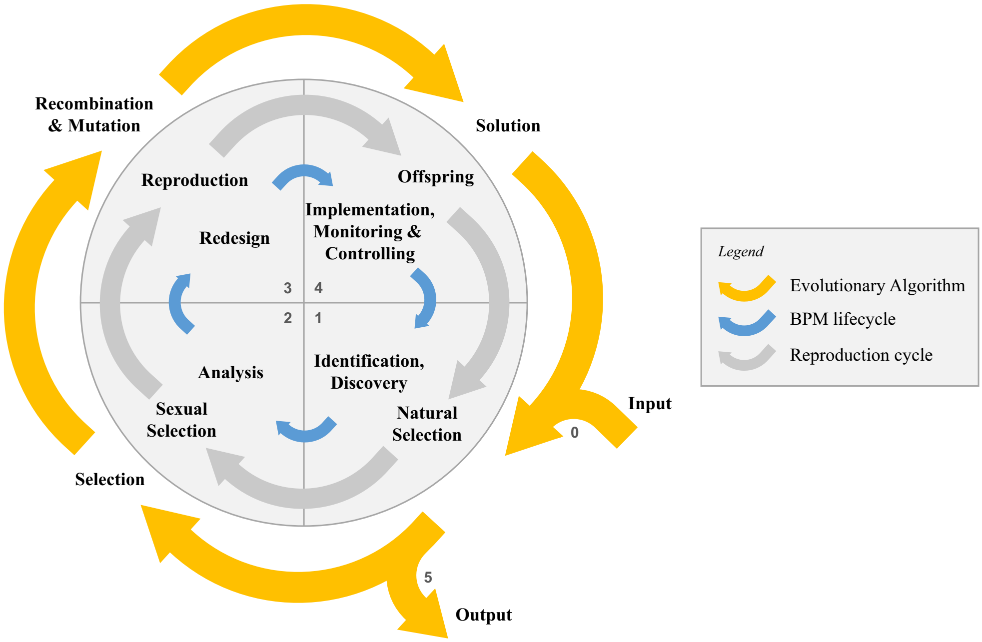


Fig. 1 Matching EA as problem solution to BPM as problem domain considering reproduction

new objects. In BPM, changes in the activities and connections as genetic information produce new, potentially better performing alternatives.

(4) Both cycles result in a new generation of objects promising better adaption to the objectives. Depending on the innovation scope, one may refer to evolution or revolution. Just like new species that may evolve in nature, the BPM lifecycle could provide new processes or business models as a radical improvement.

Basically, the BPM lifecycle and the evolutionary reproduction cycle solve an optimization problem. In doing so, BPM as a relatively new discipline could benefit from the experience of other disciplines and the related developments in CI. In the set of nature-inspired computational methodologies from CI, EAs fit best the proven improvement strategy of evolutionary principles. They draw from genetic algorithms (Holland 1992), evolutionary strategies (Rechenberg 1973; Schwefel 1977), evolutionary programming (Fogel et al. 1966), and genetic programming (Koza 1992) abstracting the evolutionary reproduction cycle (Abraham 2005). Additionally, EAs represent a suitable solution to any optimization problem in the absence of any specialized technique. They provide flexibility, adaptability, robust performance, and the ability to leave local optima. According to our theoretical reasoning, we introduce the EA approach to BPM (see outer part of Fig. 1). Thereby, the fundamental procedure of the EA is similar to the simplified procedure of the BPM lifecycle – even though the EA actually only supports the one phase of process redesign in the BPM lifecycle.

Beginning with a population of known and randomly generated objects, EAs select the best objects as “parents” for the next generations. Then, the EA recombines and mutates the selected objects following the evolutionary principles. The best objects are identified by the so called fitness function which measures the alignment of the selected objects to the overall objectives. The cycle repeats until a predefined termination and, then, returns the solutions with the highest objective value. Compared to the traditional BPM lifecycle, EAs are able to simulate many evolutionary steps at once, they are less risky and are not prone to subjective biases. Not least, first approaches in the context of BPM (Low et al. 2014; Richter-Von Hagen et al. 2005; Vergidis et al. 2012; Zhou and Chen 2003) gathered initial experience in designing the problem space and applying the mechanisms of selection, recombination, and mutation. Besides the theoretical parallels, the structural similarities of the evolutionary concepts promise to foster acceptance among practical decision-makers (see design objective O.3). Overall, we can conclude that EAs are suitable for answering the research question as they have a reasonable, theoretical underpinning for solving the redesign problem and as they are in line with our design objectives.

5.2 Translating from real-world to computational world

To provide a better understanding of the design decisions – and the constructional aspects in section 5.3 – we briefly introduce an example process. We refer to this process whenever necessary and use it for evaluation purposes in section 6.2. The example is inspired by Vergidis et al. (2007) and relates to a real travel agent process. The aim of the process is to offer holiday proposals to the customers of a travel agency: The process starts with a customer enquiry containing the relevant booking information, i.e., the travel details and the price limit. The travel agent chooses from pre-configured holiday bundles and tailors a custom proposal simultaneously. On a generic level, four process activities exist where each activity can be executed in two alternative forms. The process results in a holiday proposal and the corresponding payment details. Figure 2 sketches the design in BPMN notation.

5.3 The representation of the process components

The first and most crucial step in applying EAs to the problem domain of BPM is the solid translation of processes from real-world to computational world. According to its definition, a process or its design respectively is a combination of finite elements. Following this, process redesign becomes a NP-hard problem with a highly constrained and fragmented search space as well as many local optima (Low et al. 2014). To find and assess feasible process designs, the algorithm requires not only information about the elements but also about their characteristics. Therefore, we divide process designs into their basic elements: activities, connections, and routing decisions. In order to fulfil design objective O.1, we implement five matrices: the *activity-attribute matrix*, the *object-attribute matrix*, the *activity-input matrix*, the *activity-output matrix*, and the *activity-process-attribute matrix*.

The first two matrices describe the attributes of activities and objects. The *activity-attribute matrix* is a library of possible activities available for process redesign. The activities in the rows (represented by the variable a_x) are completely described by their functions and economic attributes in the columns. Functions describe activities on a capability level. Although activities may fulfill the same function within a process, i.e., they produce the same output, they may carry out their function differently, e.g., they may vary in the required objects. Thus, activities fulfilling the same function provide the same output with different inputs and represent alternatives. The attributes assign value contributions to activities and are required for process evaluation in later stages. As typical for VBM, we describe the efficiency of activities by expected cash flows $\mu_{a_x} = E\left[\widetilde{CF}_{a_x}\right]$ and the process risk by the variance of cash flows $\sigma^2_{a_x} = Var\left[\widetilde{CF}_{a_x}\right]$. Both distribution parameters may be gathered from historical data or expert estimates. Table 1 shows the activity list for our travel agent process with two alternative forms for each activity shown in Fig. 2, e.g., a_1 and a_2 are alternatives for the activity *Browse pre-booked packages*. In contrast to the original process from Vergidis et al. (2007) who measure the performance of activities by time and quality, we use the integrative measure of expected cash flows and added information about the variance. As Vergidis et al. (2007) did not provide this, we can easily infer the cash flows as process costs by monetizing execution times.

The *objects-attribute matrix* lists all objects that could be used during process execution. In general, most objects are used or produced by activities. However, there also exist input that is not produced by process activities (so-called process input, which is externally provided prior to execution, e.g., employees or machines) and output, which is not demanded by another activity (process output as the result of the complete process execution). In our example, the travel details and the price limit derived from the customer enquiry represent the process input, whereas the holiday proposals and the payment details are the process output. The *object-attribute matrix* denotes objects (represented by the variable o_x) in the rows as process input or output and assigns economic attributes in the columns (see Table 2). If an object is denoted as process input, the total cash outflows required for the provision of the object is the corresponding economic attribute from VBM. If an object is denoted as process output, the total cash inflows resulting from selling the output or from internal charges constitute possible economic attributes. As the process input of the travel agent process is customer information, the required cash outflows equal zero. For the process output, the travel agency charges an administration fee. Further objects, i.e., pre-booked packages and travel options, are necessary to depict a proper sequence flow. As these objects are neither

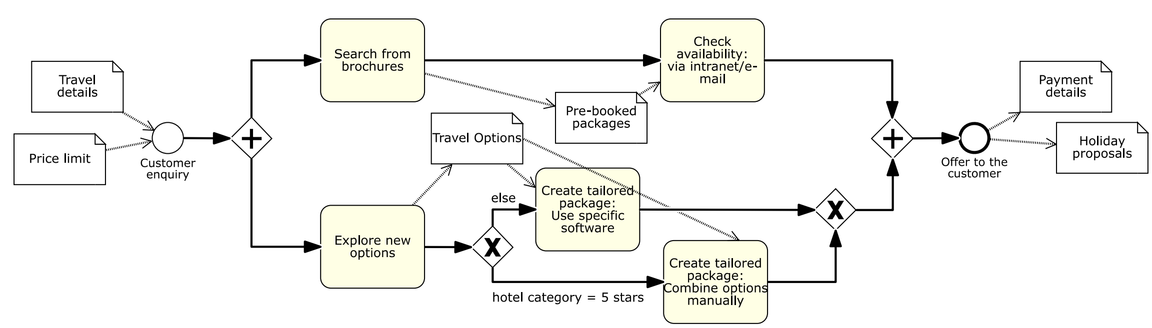


Fig. 2 Travel agent process

Table 1 Activity-attribute matrix

No.	Function	μ_{a_x}	$\sigma^2_{a_x}$
a_1	Browse pre-booked packages (PBP): Search from brochures	-1	0,30
a_2	Browse pre-booked packages (PBP): Search company intranet	-7	14,82
a_3	Explore travel options (TO): Browse past cases	-4	4,84
a_4	Explore travel options (TO): Explore new options	-23	160,02
a_5	Check availability: Via intranet/e-mail	-29	254,40
a_6	Check availability: Via phone/post	-20	121,00
a_7	Create tailored package: Use specific software	-4	4,84
a_8	Create tailored package: Combine options manually	-25	189,06

process input nor process output, they do not need economic attributes.

The other matrices describe the relationships of objects and determine the control flow of the process. The latter allows for sequential, parallel, and disjunctive executions of activities (represented by gateways in modeling notations such as BPMN). The *activity-input* and *activity-output* matrices represent the logical connectivity of activities in terms on an input-output-relationship (e.g., object o_1 is output from a_1 and input for a_2). They link the activities in the rows with the required inputs / produced outputs in the columns. This information is crucial to ensure proper object flows through the process design. According to process input and output (see *object-attribute* matrix), not all objects are both input and output in the same process design. For the chosen example, Table 3a and b show the input-output-relationships of activities and objects for the chosen example and, thus, the different alternatives for specific inputs or outputs. As a_1 and a_3 use the same input (i.e., o_1 and o_2) while creating differing outputs (i.e., o_3 for a_1 and o_4 for a_3), the information in these matrices already illustrate potential, parallel executions (e.g., both a_1 and a_3 could start at the same time when o_1 and o_2 are provided as process input). On a technical level, these matrices implement logical restrictions to our optimization problem: A process design is only feasible if the input for each activity has been provided as process or activity input in advance.

In the case of exclusive splits, routing decisions conditioned to the incoming sequence flow are required regarding which activity out of many alternatives will be executed. From a VBM perspective, conditions influence the efficiency and the risk of the process, making the implementation of execution probabilities for activities mandatory (Bolsinger et al. 2015). Focusing on data-based conditions, all process attributes known in advance or derived from execution could represent a differentiating factor. The *activity-process-attribute matrix* maps such process attributes (represented by the variable d_x) in the rows to the activities in the columns to determine under which circumstances the process is routed over a distinct activity. A process attribute is further specified by its decisive values (represented by the variable v_{x_y}) and the corresponding execution probabilities. The representation of the execution probabilities and the decisive values in turn depend on the scale of measurement of the process attribute. The matrix lists all possible decisive values and their execution probabilities. For ordinal and nominal attributes, the value range and the discrete probability distributions are entered directly. As interval scaled attributes result in continuous probability distribution, the matrix divides the value ranges into intervals and assigns the execution probabilities accordingly. In order to calculate these execution probabilities, the expected value and the standard deviation of the density function are sufficient. The distribution data may be gathered analogous to the determination of economic attributes

Table 2 Object-attribute matrix

No.	Description	Type	Process input	Process output	Price
o_1	Travel details	Information	Yes	No	0
o_2	Price limit	Information	Yes	No	0
o_3	Pre-booked Packages	Information	No	No	0
o_4	Travel options	Information	No	No	0
o_5	PBP: Holiday proposals	Information	No	Yes	20,00
o_6	PBP: Payment details	Information	No	Yes	25,00
o_7	TO: Holiday proposals	Information	No	Yes	20,00
o_8	TO: Payment details	Information	No	Yes	25,00

Table 3 Activities in relation to objects

a) Activity-input matrix

	o_1	o_2	o_3	o_4
a_1	1	1	0	0
a_2	1	1	0	0
a_3	1	1	0	0
a_4	1	1	0	0
a_5	0	0	1	0
a_6	0	0	1	0
a_7	0	0	0	1
a_8	0	0	0	1

B) Activity-output matrix

	o_3	o_4	o_5	o_6	o_7	o_8
a_1	1	0	0	0	0	0
a_2	1	0	0	0	0	0
a_3	0	1	0	0	0	0
a_4	0	1	0	0	0	0
a_5	0	0	1	1	0	0
a_6	0	0	1	1	0	0
a_7	0	0	0	0	1	1
a_8	0	0	0	0	1	1

on the basis of historical data or expert estimates. As Vergidis et al. (2007) do not consider exclusive splits, we add hotel category as process attribute for routing decisions for demonstration and evaluation purposes (see Table 4). In this case, the decisive value is the number of stars. Thus, there are five distinct attribute forms from 1-star-rating to 5-star-rating. As we assume that the relatively most hotels have a 3-star-rating, this value has the highest probability (i.e., 50 %). The effect of process attributes on the routing decisions can be shown by activities a_7 and a_8. Even though both a_7 and a_8 are two alternatives for the same activity *Create tailored package* (see Table 1) while using the same input (i.e., o_4; see Table 3a) and serving the same output (i.e., o_7 and o_8; see Table 3b), they would not be alternatives any more as both do not cover all required attribute values.

5.4 The representation of the process design

After having structured the required information about the basic elements of a redesign problem, we now elaborate the computational representation of a complete process design. As we pay attention to a communicative human-machine interface, we apply a Polish notation (also called "prefix notation") and a recursive, depth-first representation. In doing so, the processing of nested lists starts from the left hand side, similar to functional notations in MS Excel and LISP. The latter has already proven to serve many optimization problems (Koza 1992).

Following the object perspective and to ensure proper object flows, a process design always begins with the process input and ends with the process output represented by the variable *PI* or *PO* respectively. In between, the activities a_x and their logical connections describe the sequence flow. As mentioned above, these connections can have three different patterns: sequential, parallel and disjunctive. Sequences consist of two activities which have an input-output-relationship. In terms of programming, we write sequences where activity a_d follows activity a_b as an enumeration: a_b, a_d. In order to describe a parallel execution of activities, we follow a prefix notation with resemblance to the AND-function in MS Excel: $AND(a_b; a_d)$. Please note that a feasible process design requires input to execute both activities. Otherwise, the design cannot produce the desired process output. To model an exclusive split and the underlying routing decision about one out of two activities based on condition c_x, we apply the prefix XOR similar to the if-function in MS Excel: $XOR(c_x; a_b; a_d)$. The programming of conditions, in turn, requires information about the distinctive process attribute d_x and a decisive value v_{c_x} out of the possible value range from the *activity-process-attribute matrix* as well as a relational operator *r*. Technically, we use the following notation: $c_x = d_x(v_{c_x}; r_{c_x})$. To conclude a process design, we surround it with angle brackets. Table 5 summarizes the basic patterns of connections and activities our EA application is able to process.

Basically, any combination of those patterns, also nested combinations, may appear in process designs. Figure 3 provides such a complete process design based on our modified example. Starting from the left, *PI* provides the process input r_1 and r_2 for activity a_1 as well as activity a_4. Activity a_8 gets executed in process instances where the decisive characteristic "hotel category" is 5-star. For the process output, both parallel sequence flows have to be finished first. The bottom line shows the corresponding EA notification.

5.5 Customizing an EA

In the following section, we leverage the flexibility of EA algorithms. Generally, EAs benefit from the exploitative and explorative character of the underlying selection and reproduction mechanisms, making it especially appealing business problems. In order to tailor EA functionalities to our redesign problem at hand, we customize the instantiation of the initial population, apply two kinds of selection and three types of reproduction mechanisms.

6 The generation of the initial population

As proper initial populations are not biased towards areas in the problem space and approach the problem space from various directions, we compose the initial population as combinations of the status quo design and random selections of activities. The status quo design is the process as it is currently implemented and serves as a baseline for the best known solution. All other process designs created in an EA run have to compete with the status quo design as a known feasible and practicable solution. Random selections create new process designs by randomly choosing a pre-defined number of activities from the *activity-attribute matrix* to enhance the diversity of the initial population. The size of the initial population and the following generations need to be set accordingly to the

Table 4 Activity-process-attribute matrix

Process attribute	Hotel category (d_1)				
Attribute Form	★	★★	★★★	★★★★	★★★★★
Probability	2,5 %	17,5 %	50 %	25 %	5 %
	v_{1_1}	v_{1_2}	v_{1_3}	v_{1_4}	v_{1_5}
a_1	1	1	1	1	1
a_2	1	1	1	1	1
a_3	1	1	1	1	1
a_4	1	1	1	1	1
a_5	1	1	1	1	1
a_6	1	1	1	1	1
a_7	1	1	1	1	0
a_8	0	0	0	0	1

focal process. Thereby, smaller sizes have performance advantages but they more likely returns local optima. In order to illustrate our concept of initial populations, we depict an example for the travel agent process in Table 6. The population size equals 5 and the number of random activities is set equal to 4. The latter specification determines the size of the generated designs.

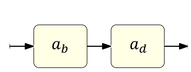
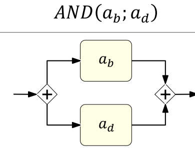
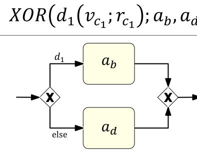
6.1 Ensuring feasible process designs by a repair mechanism

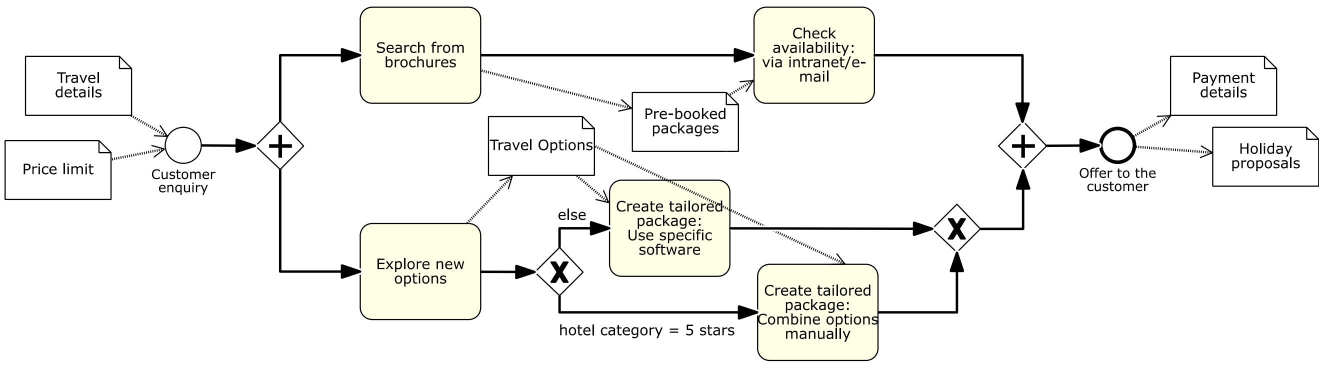
Random selections of activities rarely constitute a feasible process design, where feasibility depends on the design's ability to produce the requested process output. As infeasible solutions are less likely to provide material for producing feasible successors and as infeasible process designs will never be put into practice, we construct a repair mechanism that ensures the desired feasibility of the created solutions.

The repair mechanism operates on an activity list, e.g., the random selection of activities in the case of the initial population. It proceeds recursively and starts with the process output. If none of the activities in a design provides the process output, the repair mechanism randomly selects an activity out of the *activity-attribute matrix* that fulfils this requirement. Step by step, it determines all activities contributing to the production of the process output by either providing inputs for following activities in the object flow or by providing the process output. Besides, feasibility requires the complete coverage of present process attributes. As activities may only relate to a distinct selection of process attributes, the repair mechanism repeats these adding steps until all forms of the attributes can be processed. If a selected activity cannot get executed due to the missing input, the repair mechanism equivalently adds an appropriate activity from the library. Moreover, it erases activities that do not contribute to the production of the process output and finally returns a list of activities for a feasible process design.

Building on this master list of a feasible design, the repair mechanism arranges the activities with respect to their input-output-relationships to a process design following pre-defined rules: First, a direct input-output-relationship of activities leads to a sequence. Second, the repair mechanism arranges two or more activities using the same input and producing different output in parallel. Third, two or more activities with identical input-output-relationships but different coverages of process attributes result in an exclusive split. Thereby, the sequence flow splits with respect to all relevant process attributes. In the case of overlapping activity-process-attribute-relationships, the repair mechanism assigns the feasible activities randomly. Remaining activities not considered in any part of the sequence flow are erased as well. By applying this repair mechanism, we purely focus on feasible solutions and exploit combination patterns. Thereby, we speed up optimization and proactively exclude many misleading areas in the problem space. Limiting the problem space beforehand helps to search the remaining areas in the problem space more thoroughly and makes it more likely to determine designs with high performance.

Table 5 Basic patterns of activity combinations

Combination form	**Sequence**	**Concurrency**	**Exclusive split**
EA notation	a_b, a_d	$AND(a_b; a_d)$	$XOR(d_1(v_{c_1}; r_{c_1}); a_b, a_d)$
BPMN 2.0 notation			



EA notification	$< PI, AND(a_1, a_5; a_4, XOR(d_1(v_{1_5};=); a_8; a_7)), PO >$

Fig. 3 Example process in EA notification

Table 7 demonstrates the stepwise application of the repair mechanism to the first random selection of the initial population in Table 2. For o_7 and o_8, the repair mechanism adds the activities a_7 and a_8 going backwards from process output since the available activities do not cover all forms of the attribute "hotel category". As a_1 and a_2 or a_3 and a_4 respectively are mutual alternatives, the repair mechanism implements exclusive splits with randomly selected decisive values. Finally, the repair algorithm proceeds with arranging activities according to the pre-defined rules and creates a feasible design.

6.2 Evaluating the fitness of created process designs

In order to evaluate the potential design candidates, we follow the paradigm of VBM. More specifically, we propose the valuation function from Bolsinger (2015). This approach has four beneficial implications. First, it reduces the multi-dimensionality of the valuation problem for process redesign projects (cf. Limam Mansar et al. 2009) to a single objective which is increasing the company's value. Second, it enables the consideration of uncertainties about future process performances. Third, it extends the optimization potential of current approaches by enabling the valuation of conditions at decision nodes and integrating them into the optimization. Fourth, the application of value-based management increases the performance of EAs and enables its application also for complex processes.

Table 6 Initial Population

$< PI, AND(a_1, a_5; a_4, XOR(d_1(v_{1_5};=); a_8; a_7)), PO >$	*(Status quo)*
$< PI, a_1, a_2, a_3, a_4, PO >$	
$< PI, a_1, a_4, a_6, a_8, PO >$	
$< PI, a_1, a_3, a_4, a_7, PO >$	
$< PI, a_3, a_6, a_7, a_8, PO >$	

As one of the most accepted valuation functions, VBM proposes the preference functional ϕ. This function has proven to be applicable for decisions on the operational process level (Bolsinger 2015). The preference functional fulfills the central requirements of VBM which are the focus on cash flows, the consideration of the time value of money and of the risk attitude of the decision-maker (see design objective O.3). These requirements are fulfilled by considering three central variables: The expected net present value of process cash flows $\mu_{NPV} = E\left[\widetilde{CF}_{NPV}\right]$ as a measure of efficiency and effectiveness, the uncertainty of those cash flows represented by their expected variance $\sigma^2_{PV} = Var\left[\widetilde{CF}_{NPV}\right]$ as a measure of risk and the risk aversion of the decision-maker α. It is defined as:

$$\phi(\mu_{NPV}, \sigma_{NPV}) = \mu_{NPV} - \frac{\alpha}{2} \cdot \sigma^2{}_{NPV} \tag{1}$$

Whereas the risk aversion α is constant across process designs, our EA calculates μ_{NPV} and σ^2_{NPV} for each created process design according to Eqs. (2) and (3).

$$\mu_{NPV} = -I + \sum_{t=0}^{T} \frac{n_t \cdot \mu_p}{(1+i)} \tag{2}$$

$$\sigma^2_{NPV} = \sum_{t=0}^{T} \frac{n_t \cdot \sigma^2_p}{(1+i)^{2t}} \tag{3}$$

μ_{NPV} is defined as the difference between the initial investment for the implementation of a new process design I and the sum of the expected cash flows generated at run time. The initial investment includes a constant amount I_{fix} for conducting process redesign and a variable amount I_{var} depending on the number of new activities established. New activities lead to cash outflows for implementation and staff training among others. Within the considered time horizon T the process runs n-times in each period $t \in T$ and generates expected periodic cash flows μ_P. The periodic cash flows are then discounted by an interest rate

Table 7 Step by step guide for the repair mechanism

(1)	Check Process design for missing output:	$< PI, a_1, a_2, a_3, a_4, PO >$
(2)	Add activity that provides o_8:	$< PI, a_1, a_2, a_3, a_4, a_8, PO >$
(3)	Add further activity that provides o_8, as existing do not cover all attributes:	$< PI, a_1, a_2, a_3, a_4, a_7, a_8, PO >$
(4)	Add activity that provides o_6:	$< PI, a_1, a_2, a_3, a_4, a_5, a_7, a_8, PO >$
(5)	Repeat the steps for all other objects:	$< PI, a_1, a_2, a_3, a_4, a_5, a_7, a_8, PO >$
(6)	Erase activities that do not contribute to the production of the process output:	$< PI, a_1, a_2, a_3, a_4, a_5, a_7, a_8, PO >$
(7)	Arrange activities:	$< PI, AND(XOR(d_1(v_{1_2};=);a_1;a_2), a_5;$ $XOR(d_1(v_{1_4};=);a_3;a_4), XOR(d_1(v_{1_5};=);a_8;a_7)), PO >$

i to the present day. Similarly, we calculate σ^2_{NPV} as the sum of the variances for the single process executions σ^2_P in period t within the total planning horizon T and discount with i. General planning variables like I_{fix}, I_{var}, T, and i need to be set in advance, they do not change within an EA run and they are invariant to the process design.

In contrast, μ_P and σ^2_P are design-specific and depend on the contained activities a_x as well as their probability of appearance p_{a_x}. Equations (4) and (5) define the calculation of the economic decision variables for a process design. While an activity's expected cash flow μ_{a_x} as well as its expected standard deviation σ_{a_x} come directly from *activity-attribute matrix*, its probability p_{a_x} originates from the *activity-process-attribute-matrix* and depends on the gateways that define the paths along which a process design can be traversed.

$$\mu_p = \sum_{d=1}^{D} \mu_{ad} \cdot p_{ad} \tag{4}$$

$$\sigma^2_p = -\mu\frac{2}{p} + \sum_{d=1}^{D} \left(\sigma^2_a + \mu^2_a\right) \cdot p_{ad} + 2 \cdot \sum_{d+1}^{D-1} \sum_{b=d+1}^{D} \mu_{ad} \cdot \mu_{ad} \cdot p(a_d, a_b) \tag{5}$$

In our example, applying the repair mechanism to the initial population leads to five feasible process designs (See Table 8). The values of the fitness function with $I=0$, $T=5$, $i=2.5\%$, $n=100$, and $\alpha=0.05$ are also shown.

6.3 The selection mechanism

We apply two types of selection mechanisms: the elitist selection and the tournament selection. In the elitist selection, a defined number of currently best known designs gets directly copied to the next generation without undergoing recombination or mutation. Hence, we can ensure that the best process designs can traverse to the end. As our completing selection mechanism, we use tournament selection to balance exploration and exploitation. Thereby, we implement moderate selection pressure while still allowing for further fine tuning and preventing premature convergence towards local optima (De Jong 2006). In tournament selection, a specified number of designs of the current population competes with their fitness values $\phi(\mu_{NPV}, \sigma_{NPV})$ against each other. Thereby, the amount of competitors needs to be set in advance and remains constant throughout the optimization run. The higher the amount of competitors, the higher is the selection pressure and the more likely is premature convergence. In each competition, the design with the highest fitness value gets chosen as a parent for the next generation. For the travel agent process, Fig. 4 provides exemplary tournament selections with the winner marked in bold.

Due to a predefined recombination probability, the winning competitor is combined with a second parent from a second tournament selection into an offspring. In this case, the EA modifies the offspring additionally by the recombination and mutation mechanisms (see next section). Otherwise, the offspring is just a (probably mutated) copy of the winning competitor and not a combination of two designs. After having produced an offspring design, the parent design returns to its population and may still be a parent for further offspring. This customization enables that more than one variation of a promising design may traverse to the next generation.

6.4 The reproduction mechanisms

When creating new designs, our EA considers three mechanisms: copying, recombination and mutation. The first one, copying, retains promising process designs from the elitist selection but does not provide further information about the problem space. It ensures that the best solutions can traverse to the end. Recombination and mutation introduce new designs and, hence, help to explore the problem space. Whereas, recombination supports local search, mutation ensures global search within the problem space. Therefore, our application builds on selection mechanisms to seize designs with higher performance, it exploits recombination for combining promising designs in novel

Table 8 Fitness values of the "repaired" initial population

Process design	$\phi(\mu_{NPV}, \sigma_{NPV})$
$< PI, AND(a_1, a_5; a_4, XOR(d_1(v_{1_5}; =); a_8; a_7)), PO >$	9984.12
$< PI, AND(XOR(d_1(v_{1_2}; =); a_1; a_2), a_5; XOR(d_1(v_{1_4}; =); a_3; a_4), XOR(d_1(v_{1_5}; =); a_8; a_7)), PO >$	9420.80
$< PI, AND(a_1, a_6; a_4, XOR(d_1(v_{1_5}; =); a_8; a_7)), PO >$	15,606.79
$< PI, AND(a_1, a_5; XOR(d_1(v_{1_3}; =); a_4; a_3), XOR(d_1(v_{1_5}; =); a_8; a_7)), PO >$	14,260.88
$< PI, AND(a_2, a_6; a_3, XOR(d_1(v_{1_5}; =); a_8; a_7)), PO >$	23,166.22

ways and mutation for creating new designs. Before innovating process designs in the latter two reproduction mechanisms, our algorithm re-translates parent designs into activity lists and abstracts from the structural appearances. Thereby, we can reduce the bias towards children having the same structures and conditions in their process designs as their parents. As this condensed interpretation of recombination and mutation does not ensure that the offspring represent feasible process designs, the activity lists of the new designs undergo the repair algorithm before re-translating them into process designs.

For recombination, the parents' designs randomly exchange activities resulting in two new designs following a two-point crossover. With a predetermined probability, the first parent exchanges two of its activities for one activity (see ② in Fig. 5). Otherwise, the parents exchange one activity for another (see ① in Fig. 5). As a consequence, offspring of varying sizes evolve. For mutation, each activity in the list of the offspring is exchanged with a predetermined mutation probability against a random activity from the library (see ③ in Fig. 5). The determination of the mutation probability is crucial. A higher mutation probability leads to a higher explorative character of the EA but makes it also more similar to random search. However, if the mutation probability is low, premature convergence is likely.

6.5 Summary

The selection and reproduction mechanisms lead to offspring that, in turn, represent their parents for the next generation of process designs. This cycle will continue until a termination criterion is reached. The EA run finishes either by reaching the maximal number of generations or after a specified number of generations without a change of the best known design. Then, the EA returns the best process designs.

In all, EAs allow for a wide range of parameter settings. This flexibility enables the algorithm to cope with a high number of processes. Process designers may set the parameters according to the nature of the process at hand and their goals. Our EA shows a high exploitative character when dealing with process designs of low complexity and a higher explorative character when facing complex optimization problems. Figure 6 summarizes our results and the input parameters presented in this section.

7 Evaluation

7.1 Validation of the design specification (EVAL2)

In order to evaluate if the design specification of our computational support for process redesign suitably addresses our research question, we discuss its key features against the pre-defined design objectives obtained from justificatory knowledge. This validation corresponds to the so called feature comparison, an ex-ante and artificial evaluation method (Venable et al. 2012).

Exemplary tournament selection with number of competitors = 4
$< PI, AND\left(a_1, a_5; a_4, XOR\left(d_1\left(v_{1_5}; =\right); a_8; a_7\right)\right), PO >$
$< PI, AND\left(XOR\left(d_1\left(v_{1_2}; =\right); a_1; a_2\right), a_5; XOR\left(d_1\left(v_{1_4}; =\right); a_3; a_4\right), XOR\left(d_1\left(v_{1_5}; =\right); a_8; a_7\right)\right), PO >$
$< PI, AND\left(a_1, a_5; XOR\left(d_1\left(v_{1_3}; =\right); a_4; a_3\right), XOR\left(d_1\left(v_{1_5}; =\right); a_8; a_7\right)\right), PO >$
$< \mathbf{PI, AND}\left(\mathbf{a_2, a_6; a_3, XOR}\left(\mathbf{d_1}\left(\mathbf{v_{1_5}}; =\right); \mathbf{a_8}; \mathbf{a_7}\right)\right), \mathbf{PO} >$

Exemplary tournament selection with number of competitors = 3
$< PI, AND\left(a_1, a_5; a_4, XOR\left(d_1\left(v_{1_5}; =\right); a_8; a_7\right)\right), PO >$
$< \mathbf{PI, AND}\left(\mathbf{a_1, a_6; a_4, XOR}\left(\mathbf{d_1}\left(\mathbf{v_{1_5}}; =\right); \mathbf{a_8}; \mathbf{a_7}\right)\right), \mathbf{PO} >$
$< PI, AND\left(a_1, a_5; XOR\left(d_1\left(v_{1_3}; =\right); a_4; a_3\right), XOR\left(d_1\left(v_{1_5}; =\right); a_8; a_7\right)\right), PO >$

Exemplary tournament selection with number of competitors = 2
$< PI, AND\left(XOR\left(d_1\left(v_{1_2}; =\right); a_1; a_2\right), a_5; XOR\left(d_1\left(v_{1_4}; =\right); a_3; a_4\right), XOR\left(d_1\left(v_{1_5}; =\right); a_8; a_7\right)\right), PO >$
$< \mathbf{PI, AND}\left(\mathbf{a_2, a_6; a_3, XOR}\left(\mathbf{d_1}\left(\mathbf{v_{1_5}}; =\right); \mathbf{a_8}; \mathbf{a_7}\right)\right), \mathbf{PO} >$

Exemplary tournament selection with number of competitors = 5
$< PI, AND\left(a_1, a_5; a_4, XOR\left(d_1\left(v_{1_5}; =\right); a_8; a_7\right)\right), PO >$
$< PI, AND\left(XOR\left(d_1\left(v_{1_2}; =\right); a_1; a_2\right), a_5; XOR\left(d_1\left(v_{1_4}; =\right); a_3; a_4\right), XOR\left(d_1\left(v_{1_5}; =\right); a_8; a_7\right)\right), PO >$
$< PI, AND\left(a_1, a_6; a_4, XOR\left(d_1\left(v_{1_5}; =\right); a_8; a_7\right)\right), PO >$
$< PI, AND\left(a_1, a_5; XOR\left(d_1\left(v_{1_3}; =\right); a_4; a_3\right), XOR\left(d_1\left(v_{1_5}; =\right); a_8; a_7\right)\right), PO >$
$< \mathbf{PI, AND}\left(\mathbf{a_2, a_6; a_3, XOR}\left(\mathbf{d_1}\left(\mathbf{v_{1_5}}; =\right); \mathbf{a_8}; \mathbf{a_7}\right)\right), \mathbf{PO} >$

Fig. 4 Tournament selection examples

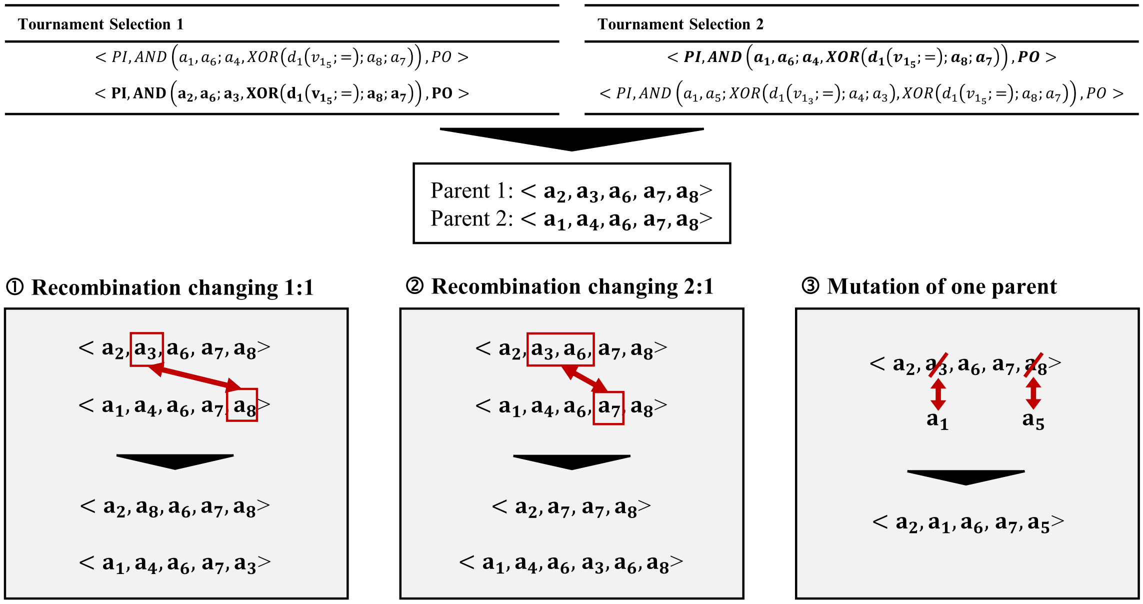


Fig. 5 Recombination and mutation examples

From a stand-alone perspective, our EA application addresses all design objectives. Table 9 illustrates details. Nevertheless, future research may improve our application with respect to some design objectives. For example, the application only considers the focal process from a stand-alone perspective and abstracts from interdependencies to other processes within the organization. An extension to a process portfolio consideration could be realized by including interdependencies in the *activity-attribute matrix*. The valuation function could then consider correlations in the variance term (O.2). Although our application computationally implements the BPM lifecycle as the most popular redesign paradigm in practice and thereby probably achieves a high acceptance among practitioners, it still remains a data-based and computational approach. A data-driven attitude and a kind of confidence into computational applications among the target users is key. Therefore, future research should investigate how our EA can be combined with more intuitive approaches like the creative redesign process (Limam Mansar et al. 2009) to further foster organizational acceptance (O.3).

7.2 Prototype construction and validation (EVAL3)

Aiming at validated artefact instantiations, we built and tested a simulation-based software prototype to provide a proof of concept. The basis of our prototype is MS Excel as it already provides basic input/output and analysis functionalities. We implemented the computational logic using VBA enabling our prototype for further applications in naturalistic settings. For computing purposes, we use a more application-friendly notation (e.g., *A01A* for a_1, *D01D* for d_1) compared to the formal EA notation.

Fig. 6 Input parameter for EA application

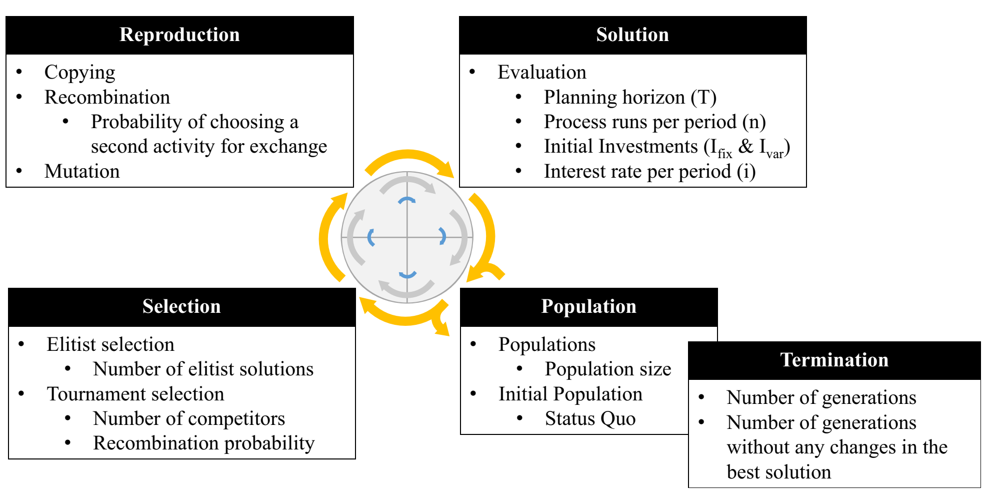


Table 9 Results of feature comparison

Design Objectives		Characteristics of our CI applications
Summary		Our algorithm supports the development of new designs that better fit restrictions of a process. It analyzes process information represented in compiled matrices, it recombines and incrementally changes activities. Finally, it prioritizes new designs with respect to their promised value contributions. Thereby, the algorithm turns the intuitive and subjective approach of "human-based" redesign initiatives to the unbiased, computational level.
(O.1)	Process Elements	Our application considers with sequential, parallel and disjunctive connections the most relevant elements from BPM. With the consideration of conditions, we can identify better designs according to process or environmental characteristics. Further, our application incrementally changes processes by a stepwise recombination of activities and connections towards a clearly prioritized set of promising designs.
(O.2)	Value-based Management	Our algorithm uses a fitness function that stems from VBM and covers cash flows, the time value of money and the risk attitude of decision-makers. The long term perspective of VBM enables us to reduce the multiple dimensions of process performance to the main economic factors of cash inflows, cash outflows and cash flow risk.
(O.3)	Evolutionary Redesign	Our algorithm is a computational implementation of the BPM lifecycle which is the most accepted redesign approach in the practical, offline world. Additionally, it deals with the most familiar design elements. The low run-time and the ability to address very complex processes further foster acceptance among practical decision-makers.

Using the prototype requires several steps. First, activities, objects, and conditions need to be defined. Second, relevant information about these elements need to be gathered to fill the five matrices: the *activity-attribute matrix*, the *object-attribute matrix*, the *activity-input matrix*, the *activity-output matrix* and the *activity-process-attribute matrix*. Third, general planning variables (e.g., planning horizon, interest rate, risk aversion) and technical EA parameters (e.g., population size, number of generations, recombination probability) need to be set. All information can be easily accessed via input spreadsheets. Several output spreadsheets summarize the results of the EA run, and provide analytic functionalities. While the EA summary sheet (Fig. 7) only lists performance information and highlights the best designs, the evaluation sheet (Fig. 8) graphically presents the development of the fitness value over generations and provides further statistics about the simulated designs as well as the included activities.

7.3 Demonstration and performance evaluation

In order to demonstrate the applicability and usefulness of our EA application, we follow a two-step evaluation. First, we apply our EA on our running example of the travel agent process (scenario A) which is based on a modified real-life scenario from Vergidis et al. (2007) to comprehensively test the correctness of our application. Second, we apply a more complex artificial setting (scenario B) to conduct further analyses.

To represent the travel agent process in the five matrices of our application, we needed to translate the performance measurement in terms of quality and time to the scale of VBM. In doing so, we used a different representation of in−/output and added information for routing decisions. Overall, the example contains eight activities where three activities have two alternatives each and where an exclusive split between activities a_7 and a_8 with respect to the chosen hotel category is mandatory. The process output consists of two objects created by two different activity sequences. Therefore, the scenario covers sequence, concurrency, and exclusive split while being simple enough to determine the optimal process design manually for comprehensively testing the correctness of the algorithm.

The EA found the best design, i.e., $< PI, AND(a_1, a_6; a_3, XOR(d_1(v_{1_5}; =); a_8; a_7))$, 44 times out of 50 independent optimization runs within the first 10 generations with 10 individual designs each. Activities a_1, a_3, and a_6 are included approximately twice as often as compared to their lower performing alternatives a_2, a_4, and a_5. Activities a_7 and a_8 are part of every

back

Start

		No. of Activities			Objective value			E[CF]			σ (E[CF])		
Run	Generatio	Average	Max	Min	Average	Max	Min	Average	Max	Min	Average	Max	Min
Run0110	G01	4,70	5,00	4,00	595,16	681,97	458,52	92,18	104,00	74,00	36,71	41,69	32,42
Run0110	G02	5,00	6,00	4,00	595,59	801,98	424,47	92,52	120,00	69,81	39,09	46,16	31,88
Run0110	G03	4,90	5,00	4,00	635,90	834,14	410,22	97,80	124,00	68,00	39,67	45,06	34,11
Run0110	G04	4,70	5,00	4,00	696,75	868,58	460,58	105,55	128,00	74,30	38,44	43,76	32,77
Run0110	G05	4,60	6,00	4,00	700,42	872,74	384,93	106,08	128,80	64,00	38,68	50,35	27,28
Run0110	G06	5,00	6,00	4,00	755,96	918,19	445,46	113,49	134,80	73,20	40,57	46,03	32,57
Run0110	G07	5,20	7,00	4,00	762,46	928,74	581,92	114,66	136,00	90,40	43,28	46,64	36,30
Run0110	G08	5,40	6,00	4,00	677,21	928,74	491,42	103,36	136,00	78,51	41,31	47,81	34,90
Run0110	G09	5,40	6,00	4,00	733,48	928,74	605,40	110,88	136,00	94,00	43,24	48,62	41,12
Run0110	G10	5,10	6,00	4,00	670,01	928,74	424,47	102,40	136,00	69,81	41,04	47,30	33,76

Fig. 7 EA summary spreadsheet

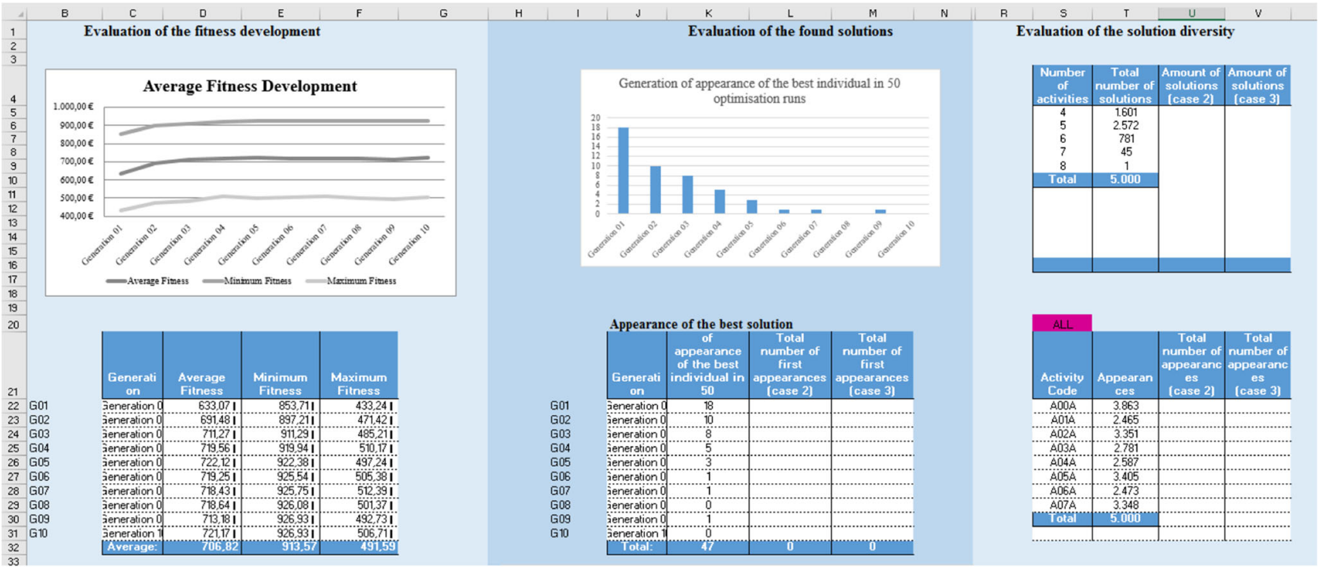


Fig. 8 Evaluation spreadsheet

solution. Due to the repair mechanism, all designs include five activities. Based on these findings, we can make several conclusions about the EA's behavior: First, the EA chooses the best alternatives if two or more activities fulfill the same functions. Second, the EA integrates conditions and exclusive splits where necessary. Third, by copying evolutionary behavior and by showing a robust performance in finding optimized designs, our EA confirms its ability as a promising tool for process redesign.

To test the EA in a more complex setting, scenario B represents challenges faced by process manager in real-world BPM problems. Accounting for a multiplicity in design options, this scenario offers different ways of transferring process input into process output as schematic shown in Fig. 9. The EA needs to combine up to nine activities according to their input-output-relationships and choose among many alternative activities (represented by the numbers attached to the activities). The alternatives vary according to their expected cash flows and uncertainty in realizing those cash flows as well as in their fit to the process attributes. The values of the economic attributes depend on the activity's function, the activity's number of sub-steps and the usage of objects and resources. Overall, the *activity-attribute matrix* contains 44 activities. Some alternatives integrate multiple sub-steps into an aggregated activity and exploit economies of scope (e.g., a_6 compared to the activity set a_3, a_4, and a_5). They are accordingly characterized by a higher efficiency (smaller expected cash outflows) compared to the sequence of the disaggregated alternatives. On the other hand, disaggregation makes the entire element easier to control and thus is exposed to lower risk than the aggregated activities. As a result, the EA also faces the trade-off between efficiency and risk. Other alternatives follow equal input-output-relationships regarding two process attributes (i.e., all activities summarized by a_{1X} and a_{2X}) to implement routing conditions at different stages of the process design. The matching of activities to the decisive forms of the process attributes results in exclusive splits just as overlapping activity-process-attribute-relationships. Summing all up, the EA faces a non-trivial problem of finding an optimal combination of activities, alternatives and routing decisions.

In 25 independent runs of 80 generations with 50 individual designs each, our EA returned the identical optimal design in more than 65 % of all cases. This design dominates all created designs as measured by the value function. Figure 10

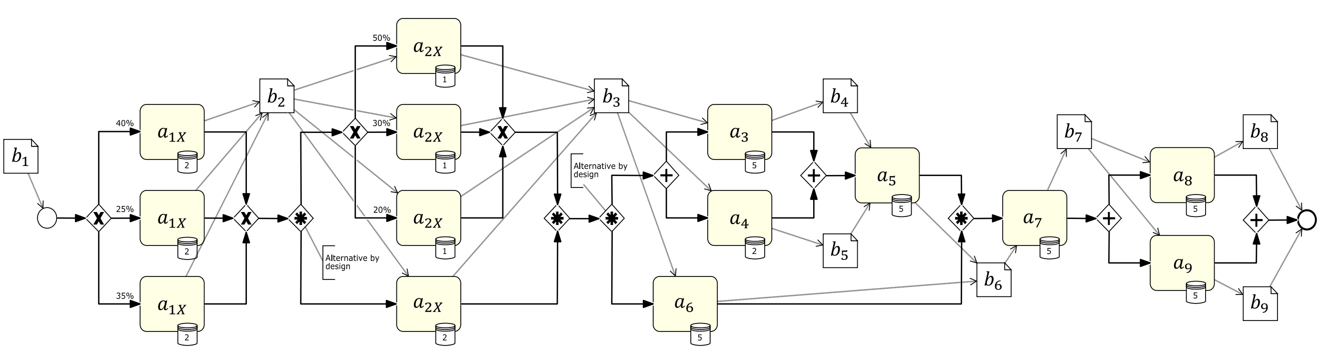
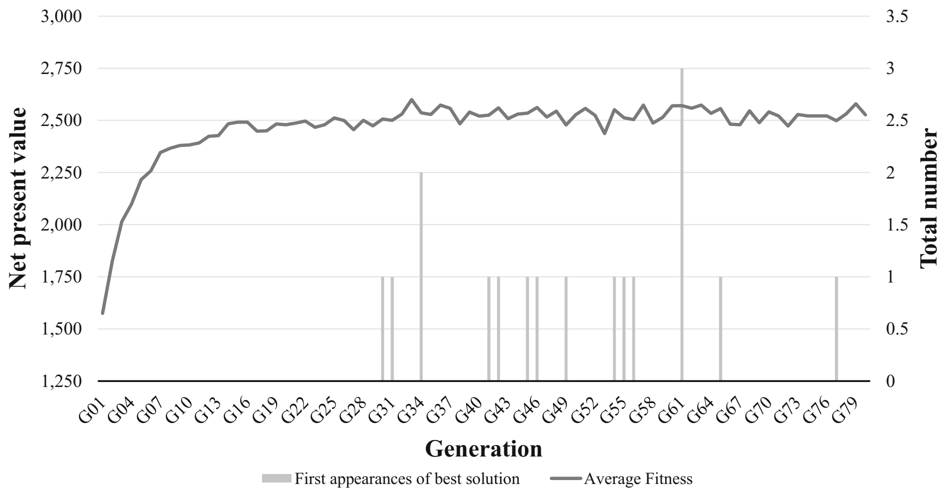


Fig. 9 Schematic representation of scenario B



Design Size	Total number
6	0
7	0
8	7.255
9	20.334
10	52.419
11	3.974
12	10.356
13	5.662
14	0
15	0

Fig. 10 Results of scenario B

provides further insights: The average fitness of progressing generations confirms the EA's exploitative character. After a high increase of the fitness at the beginning, the EA differs slightly in the designs to approach the optimal solution. This is confirmed by the distribution of the design sizes whose wide variety also illustrates the EA's explorative character. In order to find the optimized designs, the EA produced designs of six different sizes but favored designs with 10 activities. As a result of the repair mechanism, all designs include more than eight activities. The EA found the optimal design for the first time in the 30th generation.

7.4 Discussion against evaluation criteria

Further validating our prototype, we also discuss its application against typical criteria for EVAL3 as compiled and assessed by Sonnenberg and vom Brocke (2012). Summarizing, this discussion indicates that the application and the prototype address all criteria. As key findings, we can state that our approach provides an effective and efficient tool for process redesign. It builds on accessible information just as well-known representations and techniques. On the other hand, it becomes evident that applicability of our customized EA for naturalistic settings requires additional developments. Detailed results are shown in Table 10.

8 Conclusion, limitations and outlook

This paper addressed the problem how CI can support the redesign of processes. In practice, this key task of BPM often relies on human intuition and lacks the support of computational support. As a solution to this research gap, we developed an EA that incrementally improves the status quo design promising an objective basis for further discussions in a redesign committee. Following the BPM lifecycle and integrating VBM for prioritization as practice-proven and acknowledged concepts in process decision-making, our algorithm should face a high acceptance among process decision-makers as its target users. Overall, our EA unites concepts from IS research, management sciences and BPM and thereby bundles the strengths of these diverse research areas to holistically address the interdisciplinary issue of process redesign.

The main challenge in applying CI (in general) or EAs (in particular) for process redesign is the translation of process designs into the computational world. To compile the available process information, we describe activities, objects, and their logical connections as the key elements of process designs in matrices. Moreover, our algorithm is the first EA application that allows exclusive splits considering conditions based on process attributes as a further key element of processes. As a result, our EA application can develop more realistic process designs and enable a better re-translation. In order to bridge the trade-off between maintaining promising designs and searching for new solutions, the EA constructs new designs either randomly when creating the initial population or by following recombination and mutation. A repair mechanism ensures logical correctness and transforms infeasible designs, which do not produce the desired process output, into feasible designs. These feasible designs are evaluated by a valuation function from VBM and the most valuable designs form the baseline for the next generation. As a result, our algorithm can deal with complex processes in terms of a high number of activities, it provides promising design candidates in an acceptable time and it provides a clear prioritization of designs instead of a set of not-dominated designs. The entire process mimics the cognitive approach of human decision-makers but avoids the disadvantages of subjective vagueness and personal biases. It invests the strengths of CI to a real-world problem whose complexity exceeds the cognitive capacity of human beings. In other words, it constitutes a reasonable application field of human-computer interaction.

Table 10 Discussion of usefulness

Criterion	Characteristics of the CI application and the software prototype
Feasibility	The prototypical implementation and the artificial cases (scenarios A and B) illustrate that the proposed EA application is feasible for simple as well as for complex scenarios. The applied computational intelligence provides support for process redesign where other methods and mechanisms reach their limits, especially in cases of many alternative design options. Generalizing the results from our two scenarios, we can state that the EA is basically applicable to all classes of processes, but it best fits mature processes. The EA operates on diverse matrices as atomic representation of design opportunities. Accordingly, organizations need fine-grained process knowledge to apply the EA. For immature processes or young organizations, such a deep process experience could not yet have been made and filling out the process matrices is more like a blind guess. For this class of process, the unstructured redesign method as described by Limam Mansar et al. (2009) promises better results as human intuition and brainstorming methods are exploited to identify new process designs.
Ease of Use & Operationality	As we could not test our application in a real-world setting, we can only argumentatively evaluate its ease of use and operationality based on the insights we gained in the artificial environment. The EA application builds on information about activities, objects, and conditions which is already used in today's redesign initiatives. As currently conducted, the required data could be collected in automated environments by using process mining techniques. Besides, it should be possible to gather the data in non-automated environments by experts as well. The matrices for recording the data are straight forward to use as they are based on proven technologies. This argument also holds for the translation of the process designs into the computational world which we faced as greatest challenge. However, a graphical representation would assist a better understanding. As the EA should be applied repeatedly, a knowledge base should be built to institutionalize data collection routines and collect best practices.
Effectiveness, Suitability & Efficiency	The EA application can be effectively used to redesign processes. This is confirmed by the simple scenario A, which we used for plausibility checks. The fitness function as well as the repair mechanism demonstrated to ensure feasible designs. The mix of local and global search is free of subjective vagueness and uncertainty. For efficiency, we conducted performance tests with the prototype on regular work stations such as used in business environments. The EA is also highly performant in settings of various activities, objects, and conditions as well as a high amount of individual designs per generation. The optimal designs were found within a limited number of generations. In any case, the total time including recording data and applying the EA will not exceed the usual redesign time. However, simulation performance dropped from scenario A to scenario B indicating weaknesses towards the prototype's scalability.
Fidelity with real-world phenomenon	Our EA application already considers many design elements and therefore it can handle many different constellations that may occur in naturalistic settings. In particular, our inclusion of process and case characteristics as well as the ability to integrate activities and objects with different levels of detail into our computational solution provides more possibilities and flexibility towards the process design. The analogy to the BPM lifecycle allows for a minimal invasive support for process redesign. However, our application still does not consider all design elements of processes. For example, events that may occur during process execution and the corresponding waiting times are not implemented yet.
Robustness	Based on the evaluation scenarios, the EA application provides robust solutions for process redesign. In scenario A, the EA found the optimal design in all runs. In scenario B, the EA identified the same design in most instances and shows only minor deviances in the other cases, despite the risk of local optima. However, the further development should consider additional robustness checks that also cope with estimations inaccuracies, which are inevitable in naturalistic settings.

We evaluated our EA application in line with Sonnenberg and vom Brocke's (2012) framework. In this paper, we reported on the results of feature comparison, prototype construction, and demonstration examples to fulfil the requirements of the evaluation activities EVAL1 to EVAL3. As the validation revealed challenges and as our approach is beset with limitations, further research is necessary. In particular, our EA will benefit from further evaluations in real-world case studies such as recommended by evaluation activity EVAL4, where the EA and the prototype are applied in naturalistic settings. Thereby, the usefulness for organizational stakeholders involved in process redesigns could be answered in detail. Besides further evaluation, the current software prototype should also be extended towards more sophisticated visualization and analysis functionality. Thereby, it could be developed to a scalable, platform- and vendor-independent application with well-defined interfaces for data in−/output that connect to existing BPM systems. From a conceptual perspective, the growing interdependencies of processes in todays globalized times resulting in network structures necessitate adjustments to the value function. In combination with the integration of further missing process design elements (e.g., events), complexity will increase owing to this broader interpretation imposing run time and performance problems, which should also be addressed. Further research could also draw from the results of multi-criteria decision-making to enable a direct integration of other performance effects like time, quality and flexibility which we only considered implicitly.

Finally, our long-term research vision is to stepwise extend our current application until finally reaching the idealistic state of a fully computer-based BPM lifecycle. Looking at current developments regarding digitalization and big data, EAs will become even more powerful in the future. The exponential growth of available process information, e.g., gathered by WFMS, increases the potential of computational redesign as CI will get an increasing advantage over human intelligence. The cognitive capacity will become more and more deficient for the complexity of the redesign problem. To complete this outlook, the promising new designs identified by our EA could brought in a WFMS. The system could then automatically check its real-life performance and retransfer the gathered insights to the EA. Thereby, all relevant BPM activities from identifying, measuring, redesigning, and monitoring could benefit from CI in an automated cycle of improvement. Until then, our approach advances the computational tool-kit for process redesign by fusing CI, BPM and VBM to a complete application which addresses drawbacks from existing works.

References

Abraham, A. (2005). Evolutionary computation. In P. H. Sydenham & R. Thorn (Eds.), *Handbook of measuring system design*. Chichester: Wiley..

Afflerbach, P., Kastner, G., Krause, F., & Röglinger, M. (2014). The business value of process flexibility. *Business & Information Systems Engineering, 6*(4), 203–214.

Agarwal, R., & Karahanna, E. (2000). Time flies when You're having fun: cognitive absorption and beliefs about information technology usage. *MIS Quarterly, 24*(4), 665–694.

Bernstein, A., Klein, M., & Malone, T. W. (2003). The process recombinator: A tool for generating new business process ideas. 403–422. In T. W. Malone, K. Crowston, & G. A. Herman (Eds.), *Organizing business knowledge: The MIT process handbook*. Cambridge: MIT Press.

Bolsinger, M. (2015). Bringing value-based business process management to the operational process level. *Information Systems and e-Business Management, 13*(2), 355–398.

Bolsinger, M., Elsäßer, A., Helm, C., & Röglinger, M. (2015). Process improvement through economically driven routing of instances. *Business Process Management Journal, 21*(2), 353–378.

Buhl, H. U., Röglinger, M., Stöckl, S., & Braunwarth, K. S. (2011). Value orientation in process management. *Business & Information Systems Engineering, 3*(3), 163–172.

Copeland, T. E., Koller, T., & Murrin, J. (1994). Valuation: Measuring and managing the value of companies (Wiley frontiers in finance). New York: Wiley.

Damodaran, A. (2012). *Investment valuation: Tools and techniques for determining the value of any asset* (3rd ed. Wiley finance series). Hoboken: Wiley.

Danielson, M. G., Heck, J. L., & Shaffer, D. R. (2008). Shareholder theory - how opponents and proponents both get it wrong. *Journal of Applied Finance, 18*(2), 62–66.

Darwin, C. (1859). *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life* (1st ed.). London: John Murray.

Davenport, T. H. (1993). *Process innovation: Reengineering work through information technology*. Boston: Harvard Business School Press.

De Jong, K. A. (2006). *Evolutionary computation: A unified approach (a Bradford book)*. Cambridge: MIT Press.

Doomun, R., & Vunka Jungum, N. (2008). Business process modelling, simulation and reengineering: call centres. *Business Process Management Journal, 14*(6), 838–848.

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2013). *Fundamentals of business process management*. Berlin: Springer Berlin Heidelberg.

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial intelligence through simulated evolution*. New York: Wiley.

Franz, P. H., Kirchmer, M., & Rosemann, M. (2011). Value-driven business process management: Impact and benefits. https://www.accenture.com/mx-es/~/media/Accenture/Conversion-Assets/DotCom/Documents/Local/es-la/PDF2/Accenture-Value-Driven-Business-Process-Management.pdf. Accessed 16 September 2016.

Hammer, M. (2015). What is Business Process Management? In J. v. Brocke & M. Rosemann (Eds.), *Handbook on business process management 1* (2nd ed., pp. 3–16). Berlin: Springer Berlin Heidelberg.

Hammer, M., & Champy, J. (1993). *Reengineering the corporation: a manifesto for business revolution*. New York: Harper Business.

Harmon, P., & Wolf, C. (2014). The state of business process management 2014. Business Process Trends. http://www.bptrends.com/bpt/wp-content/uploads/BPTrends-State-of-BPM-Survey-Report.pdf. Accessed 3 February 2016.

Hevner, Alan R.; March, Salvatore T.; Park, Jinsoo; Ram, Sudha (2004): Design science in information systems research. In: MIS Quarterly 28 (1), S. 75–105.

Hofacker, I., & Vetschera, R. (2001). Algorithmical approaches to business process design. *Computers & Operations Research, 28*(13), 1253–1275.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence* (1st ed. Complex adaptive systems). Cambridge: MIT Press.

Ittner, C. D., & Larcker, D. F. (2001). Assessing empirical research in managerial accounting: a value-based management perspective. *Journal of Accounting and Economics, 32*(1–3), 349–410.

Keeney, R. L., & Raiffa, H. (2003). *Decisions with multiple objectives: Preferences and value tradeoffs*. Cambridge: Cambridge Univ. Press.

Kettinger, W. J., Teng, J. T., & Guha, S. (1997). Business process change: a study of methodologies, techniques, and tools. *MIS Quarterly, 21*(1), 55–98.

Kohlbacher, M., & Reijers, H. A. (2013). The effects of process-oriented organizational design on firm performance. *Business Process Management Journal, 19*(2), 245–262.

Koller, T., Goedhart, M., & Wessels, D. (2015). *Valuation: measuring and managing the value of companies* (6th ed. Wiley Finance). Hoboken: Wiley.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*(Complex adaptive systems). Cambridge: MIT Press.

Leyer, M., Heckl, D., & Moormann, J. (2015). Process Performance Measurement. In J. v. Brocke & M. Rosemann (Eds.), *Handbook on business process management 2* (2nd ed., pp. 227–241). Berlin: Springer Berlin Heidelberg.

Limam Mansar, S., Reijers, H. A., & Ounnar, F. (2009). Development of a decision-making strategy to improve the efficiency of BPR. *Expert Systems with Applications, 36*(2), 3248–3262.

Linhart, A., Manderscheid, J., & Roeglinger, M. (2015a). Roadmap to flexible service processes - a project portfolio selection and scheduling approach Germany, May 26–29, 2015. In J. Becker, J. vom Brocke, & M. de Marco (Eds.), *ECIS 2015* Proceedings, *Münster, Germany, 26.-29.5.2015* .

Linhart, A., Manderscheid, J., Roeglinger, M., & Schlott, H. (2015b). Process Improvement Roadmapping - How to Max Out Your Process. In T. Carte, A. Heinzl, & C. Urquhart (Eds.), *ICIS 2015* Proceedings, *Fort Worth, Texas, USA, 13.-16.12.2015* : Association for Information Systems.

Low, W. Z., de Weerdt, J, Wynn, M. T., ter Hofstede, A. H. M., van der Aalst, W. M., & vanden Broucke, S. (2014). Perturbing event logs to identify cost reduction opportunities: A genetic algorithm-based approach. In *CEC 2014* Proceedings, *Beijing, China* (pp. 2428–2435).

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems, 15*(4), 251–266.

March, Salvatore T.; Storey, Veda C. (2008): Design science in the information systems discipline. An introduction to the special issue on design science research. In: *MIS Quarterly* 32 (4), S. 725–730.

Mendel, G. (1866). Experiments on Plant Hybridization. In *Proceedings of the Natural History Society of Brünn* (pp. 3–47, Vol. 4). Brünn.

Min, D. M., Kim, J. R., Kim, W. C., Min, D., & Ku, S. (1996). IBRS: Intelligent bank reengineering system. *Decision Support Systems, 18*(1), 97–105.

Neiger, D., & Churilov, L. (2004a). Goal-oriented business process engineering revisited: a unifying perspective. In *ICEIS 2004* Proceedings, *Porto, Portugal* (pp. 149–163).

Neiger, D., & Churilov, L. (2004b). Goal-oriented business process modeling with EPCs and value-focused thinking. In T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, et al. (Eds.), *Business Process Management* Vol. 3080, pp. 98–115, Lecture Notes in Computer Science. Berlin: Springer Berlin Heidelberg.

Nissen, M. E. (1998). Redesigning reengineering through measurement-driven inference. *MIS Quarterly, 22*(4), 509–534.

Nissen, M. E. (2000). An intelligent tool for process redesign: manufacturing supply-chain applications. *International Journal of Flexible Manufacturing Systems, 12*(4), 321–339.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45–77.

Pries-Heje, J., Baskerville, R., & Venable, J. R. (2008). Strategies for Design Science Research Evaluation Ireland, 2008. In W. Golden, T. Acton, K. Conboy, H. van der Heijden, & V. K. Tuunainen (Eds.), *ECIS 2008* Proceedings, *Galway, Ireland* (Paper 87).

Rappaport, A. (1986). *Creating shareholder value: A guide for managers and investors*. New York: Free Press.

Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution (Problemata)* (Vol. 15). Stuttgart: Frommann-Holzboog.

Reijers, H. A., & Limam Mansar, S. (2005). Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics. *Omega, 33*(4), 283–306.

Richter-Von Hagen, C., Ratz, D., & Povalej, R. (2005). A Genetic Algorithm Approach to Self-Organizing Knowledge Intensive Processes. In *i-know '05* Proceedings, *Graz, Austria* (pp. 357–364).

Rotaru, K., Wilkin, C., Churilov, L., Neiger, D., & Ceglowski, A. (2011). Formalizing process-based risk with value-focused process engineering. *Information Systems and e-Business Management, 9*(4), 447–474.

Rutkowski, L. (2008). *Computational intelligence*. Berlin: Springer Berlin Heidelberg.

Schwefel, H.-P. (1977). *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*. Basel: Birkhäuser Basel.

Sharp, A., & McDermott, P. (2008). *Workflow modeling: Tools for process improvement and applications development* (2nd ed.). Boston: Artech House.

Siddique, N., & Adeli, H. (2013). *Computational intelligence: Synergies of fuzzy logic, neural networks, and evolutionary computing*. Chichester: Wiley.

Sonnenberg, C., & vom Brocke, J. (2012). Evaluation Patterns for Design Science Research Artefacts. In M. Helfert & B. Donnellan (Eds.), *Practical aspects of design science*, *Communications in Computer and Information Science* (Vol. 286, pp. 71–83). Berlin: Springer Berlin Heidelberg.

Stewart, G. B., & Stern, J. M. (1991). *The quest for value: A guide for senior managers*. New York: HarperBusiness.

Valiris, G., & Glykas, M. (1999). Critical review of existing BPR methodologies. *Business Process Management Journal, 5*(1), 65–86.

Van der Aalst, W. M. (2013). Business Process Management: A Comprehensive Survey. ISRN Software Engineering, Volume 2013.

Vanwersch, R. J. B., Vanderfeesten, I., Rietzschel, E., & Reijers, H. A. (2015). Improving Business Processes: Does Anybody have an Idea? In H. R. Motahari-Nezhad, J. Recker, & M. Weidlich (Eds.), *Business Process Management* (Vol. 9253, pp. 3–18, Lecture Notes in Computer Science). Cham: Springer International Publishing.

Vanwersch, R. J. B., Shahzad, K., Vanderfeesten, I., Vanhaecht, K., Grefen, P., Pintelon, L., et al. (2016). A critical evaluation and framework of business process improvement methods. *Business & Information Systems Engineering, 58*(1), 43–53.

Venable, J., Pries-Heje, J., & Baskerville, R. (2012). A Comprehensive Framework for Evaluation in Design Science Research. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, et al. (Eds.), Design science research in information systems. Advances in theory and practice (Vol. 7286, pp. 423–438, Lecture Notes in Computer Science). Berlin: Springer Berlin Heidelberg.

Vergidis, K., Tiwari, A., Majeed, B., & Roy, R. (2007). Optimisation of business process designs: an algorithmic approach with multiple objectives. *International Journal of Production Economics, 109*(1–2), 105–121.

Vergidis, K., Tiwari, A., & Majeed, B. (2008). Business process analysis and optimization: beyond reengineering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38*(1), 69–82.

Vergidis, K., Saxena, D., & Tiwari, A. (2012). An evolutionary multi-objective framework for business process optimisation. *Applied Soft Computing, 12*(8), 2638–2653.

Vom Brocke, J., & Sonnenberg, C. (2015). Value-orientation in business process management. In J. v. Brocke & M. Rosemann (Eds.), *Handbook on business process management 2* (2nd ed., pp. 101–132). Berlin: Springer Berlin Heidelberg.

Vom Brocke, J., Recker, J., & Mendling, J. (2010). Value-oriented process modeling: integrating financial perspectives into business process re-design. *Business Process Management Journal, 16*(2), 333–356.

Wastell, D. G., White, P., & Kawalek, P. (1994). A methodology for business process redesign: experiences and issues. *The Journal of Strategic Information Systems, 3*(1), 23–40.

Zellner, G. (2011). A structured evaluation of business process improvement approaches. *Business Process Management Journal, 17*(2), 203–237.

Zhou, Y., & Chen, Y. (2003). Project-oriented business process performance optimization. In *SMC '03* Proceedings, *Washington, DC, USA* (pp. 4079–4084).

Patrick Afflerbach is a research associate and a PhD student at the Research Center Finance & Information Management and the Fraunhofer Project Group for Business & Information Systems Engineering. He holds a bachelor's degree in Business Administration from the Ludwig Maximilian University, Munich, Germany, and a master's degree in Finance and Information Management from the elite graduate program of the University of Augsburg and the Technical University of Munich, Germany. Patrick's research interests include the research stream of human information behavior as well as topics from value-based business process management. He publishes or presents his research in journals or conferences, including *Business & Information Systems Engineering*, *Resources Policy*, *International Conference on Information Systems*. Patrick is also engaged in applied research projects with companies like Deutsche Bahn AG and Norddeutsche Landesbank. Furthermore, he is also involved in funded public research projects.

Martin Hohendorf is a business consultant at an international consulting firm. His current projects focus on business performance improvement. He holds a bachelor's degree and master's degree in Business Administration from the Hochschule für Technik und Wirtschaft (HTW) Berlin, Germany, or the University of Augsburg, respectively, as well as a master's degree in Leadership and Management from the Linnaeus University, Växjö, Sweden. His main research interests center around the area of business process management (BPM).

Jonas Manderscheid is a research associate at the Research Center Finance & Information Management and the Fraunhofer Project Group for Business & Information Systems Engineering. He received his doctorate in Business and Information Systems Engineering from the University of Augsburg, Germany. He studied Business Informatics (B.A.) and Information Management (M.Sc.), while also serving as (in-house) IT Consultant. Most of his work centers around business process management (BPM), focusing on the monitoring and improvement phases of the BPM lifecycle. Jonas publishes or presents his research in journals or conferences, including the *European* and the *International Conference on Information Systems,* the *International Conference on Business Process Management,* and the *Business Process Management Journal.* He is also engaged in applied research projects with companies like MLP AG and MELOS GmbH as well as funded public research projects. During his studies, Jonas dealt with BPM in the fields of wholesale, retail, and e-commerce, in addition to electronic data exchange and data analysis.