Socially-Aware Business Process Redesign

Arik Senderovich[1(✉)], Joop J. Schippers[2], and Hajo A. Reijers[3]

[1] Faculty of Information, University of Toronto, Toronto, Canada
arik.senderovich@utoronto.ca

[2] School of Economics, Utrecht University, Utrecht, The Netherlands
j.j.schippers@uu.nl

[3] Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
h.a.reijers@uu.nl

Abstract. Existing techniques for the redesign of business processes are mostly concerned with optimizing efficiency and productivity, but do not take social considerations into account. In this paper, we represent social business process redesign (SBPR) as a constrained optimization problem (COP). Assuming a workforce of human and computer resources, SBPR considers two types of decisions: (1) how to allocate tasks among this workforce and (2) which skills it should acquire. The latter decision can be used to control for the amount of automation (by setting an upper bound), which may ensure, for example, that disadvantaged workers are included. We discuss scenarios inspired by real-world considerations where the COP representation of SBPR can be used as a decision support tool. Furthermore, we present an extensive computational analysis that demonstrates the applicability of our COP-based solution to large SBPR instances, as well as a detailed analysis of the factors that influence the performance of the approach. Our work shows that it is feasible to incorporate multiple considerations into redesign decision making, while providing meaningful insights into the trade-offs involved.

1 Introduction

Socially responsible organizations look beyond shareholder interests to shape their business practices. By taking into account the interests of a broader group of stakeholders (or even society as a whole), they could become even more successful in attracting and retaining highly skilled, quality employees [1] and may enjoy a higher corporate performance than their traditional competitors [2]. The adoption of social responsibility principles in itself can also be seen as a sign of moral development of humanity, which accelerates as societies climb the stages of human empowerment [3].

In this paper, we aim to contribute to widening the conventional focus of Business Process Management (BPM) such that it can also guide and inspire socially responsible organizations. As such, it is congruent with other attempts to look beyond the traditional scope of the BPM discipline. Notably, *Green BPM* calls for a consideration of the environmental consequences of business process endeavors [4], while *Social BPM* emphasizes a better user engagement to overcome adoption issues [5].

D. Fahland et al. (Eds.): BPM 2020, LNCS 12168, pp. 75–92, 2020.
https://doi.org/10.1007/978-3-030-58666-9_5

Our particular focus is on the development of a technique that supports *process redesign*, one of the prime phases of the BPM life-cycle. This phase is eminently concerned with achieving economic benefits through its focus on efficiency and cost reduction. We introduce a novel dimension to include in redesign initiatives, namely *social responsibility*. That is, the process of decision making and process change will not only be driven by economic motives, but will also comprise social considerations. We shall refer to socially responsible redesign initiatives as Socially-aware Business Process Redesign (SBPR).

Our motivation for SBPR is rooted in a number of dilemmas that executives today face when redesigning business processes. First of all, numerous automation opportunities may exist that could be pursued to improve the efficiency of a business process. But how can automation be balanced with the social objectives of providing meaningful jobs to society and job security to employees? Secondly, executives may realize that a diverse representation of employees is righteous, social, and ethical. What is more difficult to establish is whether they can afford to train such a new workforce and how the inclusion of disadvantaged employees may affect business performance. While the management of a process-centered organization will know its processes, the activities that those processes are composed of, and the skills that are required to execute those activities, it lacks the tools to decide on how to redesign its processes while balancing productivity and social objectives.

In this work, we set to develop a decision support tool that facilitates SBPR. To this end, we formulate an SBPR problem as a constrained optimization problem (COP). The COP considers two types of decisions: (1) decisions that allocate activities to roles (classical redesign decisions) and (2) decisions on training existing or new roles to acquire new skills. The objective of our COP is to maximize efficiency, while limiting a pre-defined social budget. A *social budget* can be seen as a compensation sum, which is for example agreed upon with labor unions when a reorganization happens. The social budget is spent whenever an ‘unsocial’ decision is made, e.g., automating a task by moving it away from a human resource to a machine-based role.

Against this background, the main contribution of our work is threefold:

1. Formulating the social business process redesign (SBPR) problem as a constrained optimization problem and proving its computational complexity (Sect. 3).
2. Presenting an extensive computational analysis of the COP that shows its relevance to large SBPR instances and provides insights into problem characteristics (Sect. 4).
3. Demonstrating how decision-making with SBPR could take place in real-life settings by exploring the impact of various social policies on the associated COP (Sect. 5).

We will now first provide the background for the redesign dilemmas that we mentioned.

2 Background

In this section, we describe how recent technological developments fundamentally change the workplace. In addition, companies become increasingly aware of their social responsibilities. These elements create dilemmas for executives, which we will describe in more detail here. This section is a stepping stone towards the formulation of the optimization problem in Sect. 3.

2.1 Automation

In the past 15 years, the opinion on what human tasks can be automated has radically changed.[1] Since computers excel at following exact procedures designed by programmers, this was believed to mean that computers can only be made to perform tasks that humans have thoroughly understood and meticulously codified (see e.g. [6]).

Advances in digital technology, in particular in machine learning, are such that a much wider range of tasks are now susceptible to automation [7]. The self-driving car has become a threat to the drivers of taxis, buses, and trucks. Language translation is available to anyone with internet access. Algorithmic approaches have proved more accurate than medical specialists for a range of diagnosis tasks. Journalistic text writing can now be automated to some extent, as can personal financial advice [8]. Robotic Process Automation (RPA) is a technology that can be applied to perform some activities better and faster than human employees can [9]. So, more than ever before, *companies can improve their productivity by automating tasks hitherto performed by human workers.*

2.2 Training

Automation is, however, not the only approach to performance improvement. By investing in human capital, notably on-the-job training, workers can become more productive [10]. Training can also be used to let employees handle new technologies, e.g. AI. A third type of training is concerned with increasing the employability of people. For example, a recent analysis of German data suggests that training can be effective to move people to jobs at lower risk of automation (i.e., *requalification*) [11].

The majority of modern studies of developed economies indicate that automation and computerisation are at this point the main factors shaping the task composition of jobs. In addition, a recent study shows that the growth of non-routine cognitive tasks in Central and Eastern Europe is mostly driven by workforce up-skilling [12]. In the United States, companies like Amazon, Walmart, and AT&T just announced massive training programs for their own workers [13]. These developments show *that companies are actively looking into training as an additional way to improve their productivity.*

2.3 Inclusion

The motives of companies to invest in requalifying their existing workforce can also be explained by other than economic interests. A recent McKinsey report states that thirty-eight percent of the responding executives in a survey, coming from all world regions, cited the desire to "align with our organization's mission and values" as a key reason for initiating training programs [14]. In a similar vein, at the 2017 World Economic Forum in Davos, 80% of CEOs who were investing heavily in artificial intelligence also publicly pledged to retain and retrain existing employees (ibid). This shows that companies do realize that taking care of their workforce is a "social good".

[1] Automating a task is not the same as completely automating an *occupation* or *job*. Most human jobs involve a range of tasks.

One aspect of being a socially responsible employer is to extend hiring practices towards "nontraditional talent pools" [14]. The insight is growing that company practices to attain new employees might be biased against anyone on such bases as race, gender, sexual orientation, disability and other physical, cultural, and social attributes. An overarching concern among employers has been that the costs associated with hiring disadvantaged people, notably the disabled, will outweigh the benefits [15]. These perceived concerns with costs include the provision of expensive accommodations, decreased employee productivity, and increased supervisory time. While these are often exaggerated and a full cost-benefit analysis might also want to take into account workers' long time loyalty to the firm and the positive effects on the company's public image, an empirical study that compared a range of factors did find that the productivity (speed and accuracy) of employees with a disability are significantly lower than that of non-disabled employees [16]. In other words, *companies who want to hire responsibly, may need to account for some performance loss.*

In summary, we discussed in this section (1) that automation has become a ubiquitous instrument for productivity enhancement; (2) that training of the workforce is a further approach to productivity enhancement, with additional social benefits; (3) that other socially responsible practices, in particular the hiring of disadvantaged employees, may be costly or even negatively affect performance. This characterizes the dilemma of interest for us: how can organizations that wish to redesign their business processes balance automation, training, and hiring practices when they pursue productivity objectives as well as socially responsible outcomes? To address these questions, we need to formulate the decision problem in more precise terms, which is the focus of the next section.

3 The Problem of Social Business Process Redesign

In this section we formulate the social business process redesign (SBPR) problem as a constrained optimization problem (COP). The COP can then be used as a decision support tool for social redesign. As a running example, we shall consider the automation of the outpatient clinic process described in Fig. 1. Patients arrive at the front desk and register with the clerk. Next, their vital signs and basic lab tests are collected by the nurse. The nurse then sequences the patients according to their level of urgency and sends them to the medical doctor, who examines and treats each patient. After treatment, patients continue for a check-out with the clerk. We aim at redesigning the process such that several activities will be allocated to an automated resource.

In what follows, we start by presenting the input parameters of the SBPR problem, followed by a definition of the decision variables and the COP formulation. Subsequently, we apply the approach to our running example and conclude the section with a discussion on setting SBPR parameters.

3.1 Input Parameters

We start by describing the input parameters to the SBPR problem. Let $\mathcal{A}$ be the set of activities to be (re-)allocated to a set of resource types $\mathcal{R}$. In the as-is model presented in Fig. 1 there are 7 activities performed by 3 resource types, which we denote

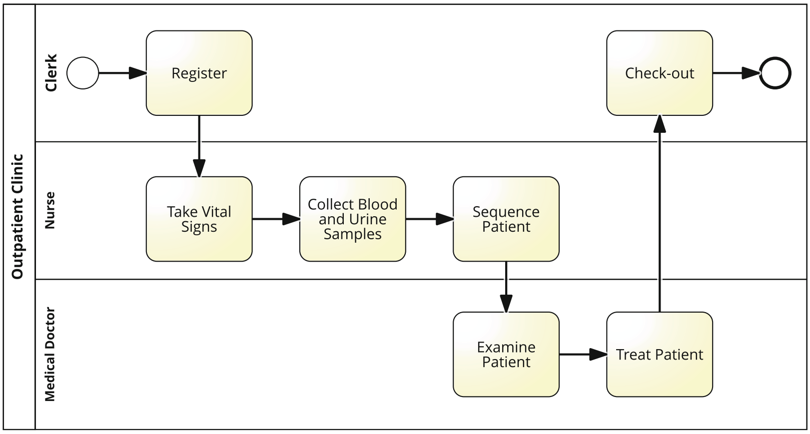
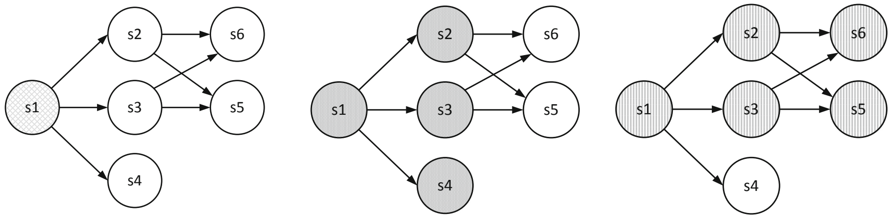


Fig. 1. An outpatient hospital treatment process.

by $C, N, M \in \mathcal{R}$ for Clerk, Nurse, and Medical Doctor, respectively. In our model, a resource type $r \in \mathcal{R}$ can execute an activity $a \in \mathcal{A}$ if and only if the resource type possesses the skill required to execute the activity. Returning to the process in Fig. 1, resource type 'Clerk' must possess the relevant skill 'Clerkship', which is required for executing 'Register' and 'Check-out'. Formally, we denote by $\mathcal{S}$ the set of skills that resource types are required to learn in order to perform the various activities. In our running example, we consider 6 skills, namely 'Clerkship', 'Vital Signs', 'Lab Tests', 'Patient Sequencing', 'Examine', and 'Treat', which we denote by $s_1, \ldots, s_6$, respectively (e.g., s_1 denotes 'Clerkship'). We assume that an activity a requires exactly one skill, which we denote by $s(a) \in \mathcal{S}$. For example, activity 'Take Vital Signs' requires s_2, which is the 'Vital Signs' skills.

We model skill acquisition using a directed acyclic skill graph $\mathcal{G}(\mathcal{S}, E)$ with its vertices being skills and its edges $E \subseteq \mathcal{S} \times \mathcal{S}$ corresponding to precedence relation between skills. For example, an edge $(s_1, s_2) \in E$ implies that one must acquire skill s_1 prior to acquiring skill s_2. By definition, a skill may have more than a single predecessor. Furthermore, we assume that a single universal skill graph exists, meaning that the clerk can learn skills that are currently possessed by nurses. We assume that the skill graph is given to us as an input. Note that our definition of a skill graph is inspired by *career paths graphs* defined in [17]. In practice, one can elicit a skill graph using existing documentation and other types of organisational data.

Figure 2 demonstrates a possible skill graph that corresponds to our running example. Note that the graph in all three figures remains the same, while the skills possessed by the different resource types are different. The as-is set of skills possessed by the resource types is defined using a coloring function $\sigma : \mathcal{R} \rightarrow 2^{\mathcal{S}}$ that maps resource types to their current sets of skills. The three different coloring functions presented in Fig. 2 correspond to the current skills of the three resource types in our running example. Note that since $(s_1, s_2) \in E$ we get that $s_1 \in \sigma(N)$, which means that a nurse can



(a) Clerk's coloring function (b) Nurse's coloring function (c) Medical Doctor's coloring function

Fig. 2. Skills graph with 3 different coloring functions of the three resource types

also perform clerkship-related activities. The as-is coloring function of every resource is assumed to be known.

3.2 Decision Variables

Having defined the inputs to the problem, we are now ready to introduce the decision variables of SBPR. The first decision that we must make in order to solve the redesign problem is allocating activities to resource types. We denote by $x_{a,r} \in \{0, 1\}$ the decision variable that equals to 1 if activity a is assigned to resource type r. This is a 'classical' redesign decision, which must be considered in any BPR initiative. In this work, we assume that an activity must be allocated to exactly one resource type. The cost of allocating activity a to resource type r is denoted by $w_{a,r}$. In practice, these costs may correspond to full-time equivalent (FTE) number of resource type r (per year) that we require to perform activity a. The quantity can be scaled by the wages of the different resource types.

Another decision that we allow in SBPR is for resource types to learn new skills. Formally, we denote by $y_{s,r} \in \{0, 1\}$ the decision variable of whether resource type r acquires skill s. Skill acquisition is associated with a learning cost $l_{s,r}$, which corresponds to various expenses related to training, hiring, and programming (in case the new resource type is a computer). In the running example, we may decide that nurses should be up-skilled to perform examinations and treatments. This skill acquisition may be expensive, yet it may pay off overall due to savings in activity allocation costs. The learning costs can also be used to specify that not all skills can be acquired by each resource type. Note that one of the strengths of our SBPR formulation is the symmetric treatment of human resources and computers. Both are treated as resource types that can be allocated to activities and trained to acquire new skills.

3.3 Objective Function and Constraints

To represent the aforementioned trade-offs between the two types of costs (activity allocation and learning), the objective function of our redesign problem minimizes the following expression:

$$\min_{x,y} \sum_{a\in\mathcal{A}} \sum_{r\in\mathcal{R}} w_{a,r} x_{a,r} + \sum_{r\in\mathcal{R}} \sum_{s\in\mathcal{S}} l_{s,r} y_{s,r}. \quad (1)$$

The first term represents the total costs of assigning activities to resource types. The second term corresponds to the total cost of resource type r learning skill s. We assume that each activity must be assigned to exactly one resource type. This corresponds to a set of constraints, $\sum_{r\in\mathcal{R}} x_{a,r} = 1, \forall a \in \mathcal{A}$. Since we aim at *social* redesign, we define the social cost that we incur when assigning activity a to resource type r as $c_{a,r}$ and assume that a social budget b_r is set by the organization for every resource type r. The budget is an upper bound on the total cost of 'unsocial' decisions made with respect to resource type r. For example, if resource A is an RPA tool, allocating many activities to A will result in high usage of the social budget. In practice, the social costs and budgets are based on a company's social policy. Companies that aim at limiting the scale of automation would assign higher costs to automating certain activities, while setting lower social budgets for computerised resources. Furthermore, organizations that target inclusion would assign higher social costs and lower social budgets to advantaged resources, compared to their disadvantaged colleagues. To represent the relation between social costs and social budgets, we add the following set of constraints to our redesign problem:

$$\sum_{a\in\mathcal{A}} c_{a,r} x_{a,r} \leq b_r, \forall r \in \mathcal{R}. \quad (2)$$

In addition, for each activity a and resource type r for which $x_{a,r} = 1$, either $s(a)$ is already in the existing skills of the resource type, i.e., $s(a) \in \sigma(r)$, or we train r to obtain skill $s(a)$. Formally, we add the constraints:

$$(x_{a,r} = 1 \wedge s(a) \notin \sigma(r)) \rightarrow (y_{s(a),r} = 1), \forall a \in \mathcal{A}, r \in \mathcal{R}. \quad (3)$$

Lastly, a skill s can be obtained only if all its predecessor skills $s' : (s', s) \in E$ were obtained. This yields the following constraints:

$$y_{s,r} = 1 \;\rightarrow\; \forall(s', s) \in E(y_{s',r} = 1 \vee s' \in \sigma(r)), \forall s \in \mathcal{S}, r \in \mathcal{R}. \quad (4)$$

Given the above objective function and the set of constraints, the SBPR problem can be written as the following constrained optimization problem (COP):

$$\begin{aligned}
\min_{x,y} \quad & \sum_{a\in\mathcal{A}} \sum_{r\in\mathcal{R}} w_{a,r} x_{a,r} + \sum_{r\in\mathcal{R}} \sum_{s\in\mathcal{S}} l_{s,r} y_{s,r} & \\
\text{s.t.} \quad & \sum_{a\in\mathcal{A}} c_{a,r} x_{a,r} \leq b_r, & \forall r \in \mathcal{R} \\
& \sum_{r\in\mathcal{R}} x_{a,r} = 1 & \forall a \in \mathcal{A} \\
& (x_{a,r} = 1 \wedge s(a) \notin \sigma(r)) \rightarrow (y_{s(a),r} = 1) & \forall a \in \mathcal{A}, r \in \mathcal{R}, \\
& y_{s,r} = 1 \;\rightarrow\; \forall(s', s) \in E\ (y_{s',r} = 1 \vee s' \in \sigma(r)) & \forall s \in \mathcal{S}, r \in \mathcal{R}, \\
& x_{a,r} \in \{0,1\}, y_{s,r} \in \{0,1\} & \forall s \in \mathcal{S}, \forall r \in \mathcal{R}, \forall a \in \mathcal{A}.
\end{aligned} \quad (5)$$

The COP in Eq. (5) can be solved using standard constraint solvers. The following result states the computational complexity of the SBPR.

Theorem 1. *The SBPR problem defined in Eq. (5) is $\mathcal{NP}$-complete.*

Proof. We show by reduction into the *generalized assignment problem* (GAP), which is known to be $\mathcal{NP}$-hard [18]. For a special case of the problem when $\forall s, r(l_{s,r} = 0 \wedge s \in \sigma(r))$, i.e., each learning cost is zero and each resource type has all skills, we get that the objective function comprises only the first expression and the two implication constraints are satisfied. The latter stems from the fact that the first implication

$$(x_{a,r} = 1 \wedge s(a) \notin \sigma(r)) \rightarrow (y_{s(a),r} = 1)$$

always holds, since $s(a) \in \sigma(r)$ for any activity. The second implication has a true right-hand side regardless whether $y_{s,r} = 1$ or not. Removing the two implication constraints and the second objective term turns the problem into an instance of the GAP [18]. Hence, we get that the GAP is a special case of SBPR, which makes SBPR at least as hard as the GAP, namely at least $\mathcal{NP}$-hard. Since the SBPR can be formulated as a mixed-integer programming using the constraint reformulation in [19], we get that SBPR's computational complexity is at most $\mathcal{NP}$-hard. Therefore, the complexity of SBPR is $\mathcal{NP}$-complete. □

This appears to be a discouraging result for the general formulation of the SBPR. However, our experiments show that for some variants of large problems (1000 activities, 500 resource types, 100 skills) the run time is a matter of seconds when using a constrained solver. In our computational analysis of SBPR (Sect. 4) we pinpoint the conditions that make the problem intractable.

3.4 Applying SBPR to the Running Example

To show how SBPR can be applied in practice, we instantiate the running example in Fig. 1. We use the activities and resources as depicted in the BPMN diagram. Moreover, we consider the skill graph in Fig. 2. Next, we add a new resource type, which is an RPA tool that can be trained to perform clerkship (acquire skill s_1). Therefore, the new resource set is now $\mathcal{R} = \{C, N, M, A\}$ with C, N, M being Clerk, Nurse, and Medical Doctor, as before, and A being the RPA solution. The activities that must be allocated remain as before. We set the parameters of the model as follows:

- The weights $w_{a,r}$ are set such that the RPA tool receives $w_{a,A} = 0$ for each activity (assuming that once the tool is trained and deployed, its costs are 0). The other weights are set according to the yearly salary that the human resources receive. We assume that weights are dependent only on resources (and not the activities) and set $w_{a,C} = 1, w_{a,N} = 3, w_{a,M} = 9, \forall a \in \mathcal{A}$. Note that we do not assume that some of the activities cannot be automated.
- The social cost $c_{a,r}$ is set to be 0 for human resource types and $c_{a,A} = 1$ for the proposed RPA solution.

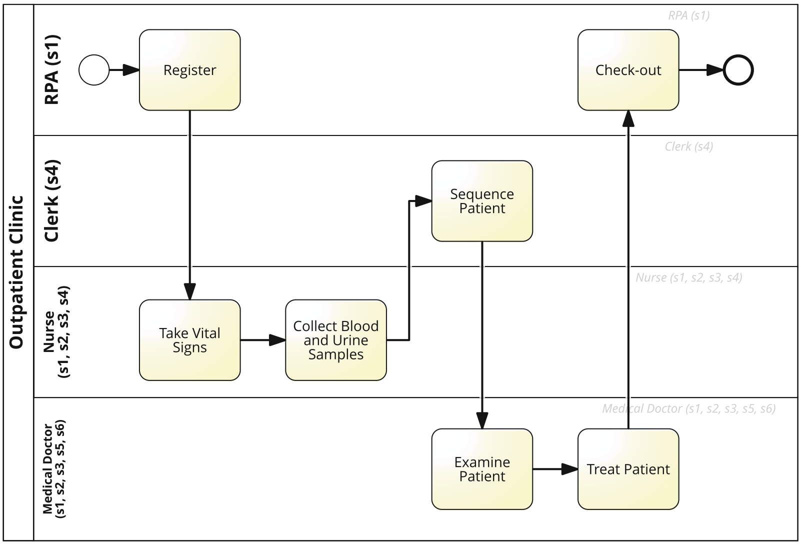


Fig. 3. SBPR result for the outpatient hospital treatment process.

– The social budget is set to be equal to the number of activities for human resources ($b_r = 7$), since we do not wish to limit the number of activities that they can perform. On the other hand, we set $b_r = 2$ for our RPA tool, implying that we allow computerisation of at most two activities.
– The learning costs $l_{s,r}$ are set to be high for human resource types ($l_{s,r} = 10000$), except for the clerk who is allowed to up-skill and learn the sequencing of patients ($l_{C,s_4} = 1$). In alternative scenarios, one may wish to up-skill nurses to perform some of the medical doctor's activities. For the RPA tool, A, we set the learning cost of clerkship and sequencing to be low, $l_{A,s_1}, l_{A,s_4} = 1$, while setting the costs to learn other skills (e.g., 'Treat' and 'Examine') to be high ($l_{A,s} = 10000, \forall s \neq s_1, s_4$).

Figure 3 presents the resulting redesigned process model that stems from a solution of the COP. Note that the acquired skills are embedded into the name of the corresponding resource types for each lane. We observe that the clerk resource type was trained to perform the 'Sequence Patient' activity, while the RPA tool was trained for the 'Register' and 'Check-out' activities. The 'Sequence Patient' activity was not chosen to be computerised, since performing 3 activities is outside the social budget of the RPA. Without a social element in the re-design initiative, the clerks would remain without any activity assignments and their role would become obsolete.

Lastly, note that the as-is solution is also a feasible solution to the problem. However, it is suboptimal, since it yields an objective value of 29, while the optimal value is 27. The difference of 2 units may well be substantial if we consider a unit to be the wage of a full-time equivalent position. Without the social consideration, we could achieve an objective value of 26, which stems from the infeasible solution of automating patient sequencing.

3.5 Setting SBPR Inputs

In this part, we generalize from a specific application of our approach to setting SBPR input parameters in realistic settings. We start by an observation that a key distinction in how we set the various input parameters comes from their origin.

The set of weights, $w_{a,r}$, the learning costs $l_{r,s}$, and the skill graph (including the as-is coloring functions) are *exogenous* to the SBPR problem. These exogenous parameters can be estimated using organizational data. For example, the weights $w_{a,r}$ can be computed as the number of FTEs of resource type $r \in \mathcal{R}$ that were historically required to perform activity $a \in \mathcal{A}$. Similarly, one can assess past costs of training using the total manpower required to acquire a skill s for resource type r. The skill graph and the coloring functions can be derived from employee training guidelines, professional curricula, and other organisational documents.

Conversely, social costs and resource budgets are *endogenous*, since they represent organisational policies concerning social responsibility. Clearly, by setting all social costs $c_{a,r}$ to be 0, an organization would be stating that they do not wish to be socially responsible for the distribution of work. The SBPR would then collapse into a simple task to resource allocation problem. We shall demonstrate the implications of setting different social policies on the corresponding SBPR implementations in Sect. 5, but will perform a computation analysis of the SBPR first.

4 Computational Analysis of SBPR

In this part, we describe a thorough computational analysis, which we conducted using synthetically generated instances of SBPR. To demonstrate the applicability of the COP, we measure the run-time of solving SBPR to optimality as function of various controlled variables (e.g., number of activities, number of skills, and ratio between activities and resources). We shall first describe our experimental design, followed by the main results and insights gathered from the evaluation.

4.1 Experimental Design

In this part, we discuss the experimental setting that we used for our empirical analysis. Below, we provide the methods we used to generate input parameters, control for the computational complexity of the SBPR problem, the controlled and uncontrolled variables, and details on the implementation of our approach.

Generating Input Parameters. In Theorem 1, we proved that the GAP is a special case of SBPR. Hence, for our experiment we used well-established GAP problem instances to create instances of SBPR. Specifically, we generated sets of parameters from previous work that analyzed the GAP's computational complexity [20]:

– Allocation coefficients $w_{a,r}$ were sampled from a discrete uniform distribution $U[15, 25]$,

– Social cost coefficients $c_{a,r}$ were sampled from $U[0, 25]$, thus allowing for some activities to have 0 social cost (i.e., they can be fully automated).

In addition, we created skills sets of sizes $10, 50, 100$, randomly assigning the skills required to perform the activities by setting $s(a), \forall a \in \mathcal{A}$. We created random skill graphs with random initial coloring functions (σ) and sampled learning coefficients ($l_{s,r}$) from uniform distribution $U[5, 25]$.

Controlling for Computational Complexity. In order to control for the hardness of the SBPR (as it depends on the hardness of the GAP), we have used a well-known result that the computational complexity of the GAP depends mainly on the ratio between the left-hand side (LHS) and the right-hand side (RHS) of the constraints in Eq. (2) [20]. Specifically, the LHS expression, $\sum_{a\in\mathcal{A}} c_{a,r}x_{a,r}$, corresponds to the demand for the budget expressed in the RHS. We shall refer to the LHS as the *social pressure* on a resource. One can show that the GAP becomes hard when the mean total social pressure approaches the total social budget [20], i.e., $\frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}}\sum_{a\in\mathcal{A}} c_{a,r} \approx \sum_{r\in\mathcal{R}} b_r$. In GAP experiments, one typically sets the budget b_r using a parameter ρ by setting:

$$b_r = \frac{\rho}{|\mathcal{R}|}\sum_{a\in\mathcal{A}} c_{a,r}. \tag{6}$$

When ρ decreases, the social pressure per resource increases (and with it the computational complexity of the GAP), and vice versa. Therefore, ρ can be thought of as the *inverse* social pressure. In the literature, ρ is often set to be 0.8, which is a value known to be generating hard instances. Similarly, to control for the hardness of the SBPR problem, we vary the values of ρ between 0.5 and 0.99. SBPR problems with $\rho < 0.5$ were often found to be infeasible. So, in the final experiments we used 0.5 as the lowest value for ρ.

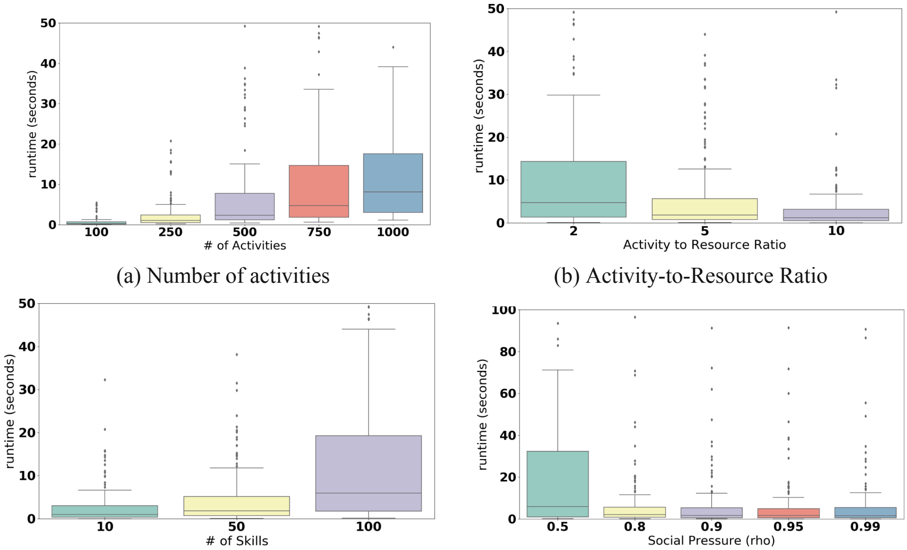
Controlled and Uncontrolled Variables. Below, we summarize the values of the controlled variables in the randomly generated SBPR instances:

– The number of activities $|\mathcal{A}|$ varied in $\{100, 250, 500, 750, 1000\}$,
– Activity to resource type ratio $\frac{|\mathcal{A}|}{|\mathcal{R}|}$ was set to be $2, 5, 10$ (with 2 meaning that there are 2 times more activities than resource types),
– The number of skills $|\mathcal{S}|$ was set to be in $\{10, 50, 100\}$,
– Social pressure per resource, ρ, received values in $\{0.5, 0.8, 0.9, 0.95, 0.99\}$.

The uncontrolled variable was the run-time (in seconds), which is defined as the time until a feasible solution was proven to be optimal. The experimental setting led to 675 randomly generated SBPR instances, which served as the data points for the statistical analysis provided in Sect. 4.2.

Implementation Details. We implemented the SBPR problem using Minizinc [21], a constraint modeling language that works with a plethora of solvers. The batch optimization of all instances was conducted using the *pymzn* Minizinc wrapper for Python[2]. The experiments were conducted on a Dell Inspiron Intel machine with i7-8565U CPU @

[2] http://paolodragone.com/pymzn/.



(a) Number of activities

(b) Activity-to-Resource Ratio

(c) Number of skills

(d) Social Pressure

Fig. 4. Run-time as function of the main effects.

1.80 GHZ, 16 GB of RAM, and 512 GB of SSD external memory. The problem definition in Minizinc and the code that generates instances and solves them to optimality are available online[3].

4.2 Main Results and Empirical Insights

We treat the controlled variables as categorical factors that influence the run-time. We performed an analysis of variance (ANOVA) for the run-time across the four controlled factors, namely the number of activities, the activity-to-resource ratio, the number of skills, and the inverse social pressure. The main effects of the controlled variables and the interactions between those variables were found to have a statistically significant influence on the run-time. Below, we provide a graphical exploration of both the main effects and significant three-way interactions.

The four box plots in Fig. 4 present the run-time as function of the main effects. The lines crossing the box plots correspond to the median run-time per level and box limits correspond to the 5th and the 95th percentiles, respectively. According to the main effects, the run-time grows with the number of activities and the number of skills; it decreases as the activity-to-resource ratio increases and as the social pressure becomes smaller (ρ increases).

We turn to present two significant interactions between our controlled variables. We first examine the three-way interaction between social pressure, activities, and ratios. Figure 5 presents an interaction plot. The points represent the median values, while the

[3] https://bit.ly/2Q2H18R.

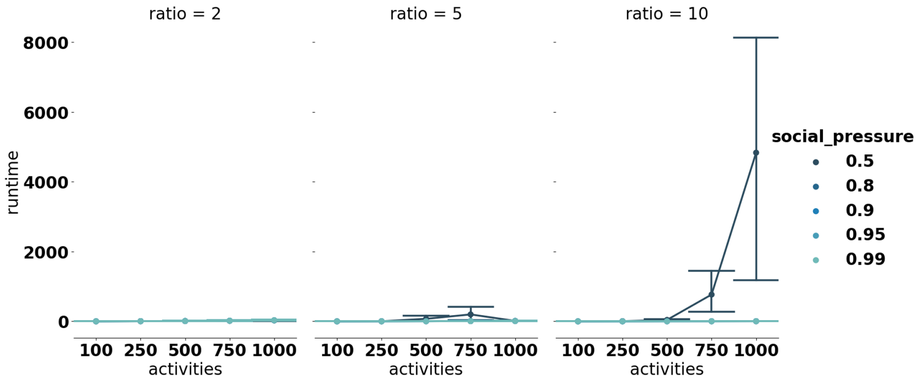


Fig. 5. Three-way interaction: activities, social pressure and ratio.

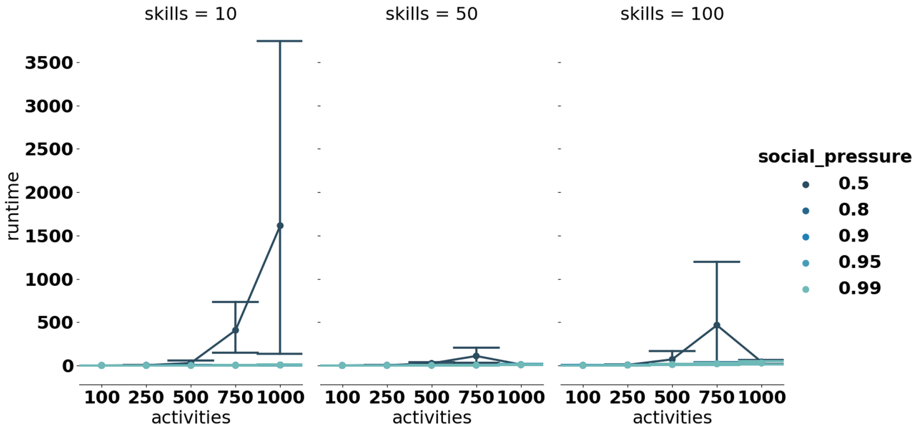


Fig. 6. Three-way interaction: activities, social pressure and skills.

intervals correspond to the upper and lower 5% quantiles, as in Fig. 4. Surprisingly, we observe that the run-time does not increase significantly as function of the number of activities nor the social pressure, as long as the activity-to-resource ratio is 2 or 5. When the ratio becomes 10, we have scarce resources and run-time increases exponentially. Therefore, unlike what we see in the main effect of activity-to-resource ratio, the computational complexity increases exponentially with the ratio when the social pressure is kept high ($\rho = 0.5$).

This phenomenon can be explained by reduced flexibility when assigning jobs to a scarce amount of resource types with limited social budgets. For example, if an RPA tool has met its budget due to high social pressure, having less alternative human resources leaves less options to distribute the remaining social pressure. Conversely, having more resource types available (per activity), turns the problem of finding a different allocation for the activities into an easy one, regardless of the absolute number of activities.

Next, we continue by presenting the three-way interaction between the number of skills, the number of activities, and the social pressure (Fig. 6). We observe that as the numbers of activities grow and with the increase of social pressure, having less skills increases the run-time exponentially. In other words, when less skills are present, resources are limited in learning and re-qualifying. Therefore, when the social pressure and the number of activities per resource increase, we cannot use learning as an alternative solution to allocating activities. This leads to a higher computational complexity, essentially turning the SBPR problem into a GAP.

These observations imply that in settings where the social pressure is high compared to the social budget, the approach is efficient only in with high number of skills and low activity-to-resource ratios. Otherwise, using the COP to find optimal allocation and re-qualification decisions can become impractical for large instances. This may be partly countered by choosing a different level of granularity of skills and resource types when using the COP.

5 SBPR-Based Decision Making

In this part, we present considerations related to social policies of organisations that wish to apply our approach. Specifically, we discuss the question *how different social policies would influence SBPR-based decision making?* To answer the question, we provide three settings in which we demonstrate how different organizational policies impact the resulting SBPR problems and their corresponding redesign solutions.

Across all settings, we shall assume the existence of two special resource types, $m, d \in \mathcal{R}$, that correspond to a computerised resource type m (e.g., an RPA tool) and a disadvantaged employee group d [4]. We shall denote the set of all other resource types by $\mathcal{R}^-$, i.e., $\mathcal{R}^- = \mathcal{R} \setminus \{m, d\}$. We shall assume that $|\mathcal{R}^-| > 0$, which implies that there is at least a single advantaged human resource type. For the analysis, we shall assume that the weights are ordered as follows:

$$w_{a,m} < w_{a,r} < w_{a,d}, \forall r \in \mathcal{R}^-, \forall a \in \mathcal{A},$$

which implies that computerisation is always more economic to implement than to hire humans, and that inclusion is always less beneficial (from a strictly short-term, economic perspective) than hiring advantaged humans.

Setting 1: Limiting Automation. Learning Costs are Negligible. In this setting, we assume that the learning costs are negligible compared to the operational weights $w_{a,r}$, i.e., $l_{s,r} = 0, \forall s, r$. Furthermore, we assume that the decision makers only strive to limit automation. Therefore, they set $c_{a,m} = 1$ and $c_{a,r} = 0, \forall r \neq m$. The automation budget is set to be $|\mathcal{A}| > b_m > 0$ while the budget for human resource types $b_r, \forall r \neq m$ is set large enough to be considered infinite. Since the learning costs are negligible, we can train any resource type to perform any activity. Hence, there is always a feasible solution, regardless of the initial coloring functions (we can always train one of the

[4] One can easily generalize the applicability analysis that follows to sets of M automation types and D disadvantaged employee types.

human resource types to perform all activities). Since $w_{a,m} < w_{a,r}, \forall r \neq m$, an optimal solution will always allocate exactly b_m of the activities to resource m. Moreover, it will allocate to m the first b_m activities with respect to an order by $w_{a,r} - w_{a,m}$. Lastly, the solution will assign the rest of the activities to advantaged human resources ($r \neq d$), since $w_{a,d} > w_{a,r}, r \neq d$.

In this setting, the social policy neglects inclusion, because the social cost of assigning an activity to *any* human resource is set to 0.

Setting 2: Still Limiting Only Automation. Learning Costs are Non-negligible. As a policy, the organization is only limiting automation by setting the social costs and budgets as described in Setting 1. Since learning costs are non-zero, it may be beneficial to train disadvantaged resources and assign them to newly learned tasks, compared to learning new skills and assigning activities to advantaged human resource types (e.g., due to a government program)[5]. In this setting, we may achieve inclusion indirectly, depending on the values of exogenous parameters.

Setting 3: Automation and Inclusion via Training. In the last setting, we assume that policy makers aim at both inclusion and automation. By setting social costs and budgets properly, it can be guaranteed that for any values of exogenous parameters that allow training of disadvantaged employees, at least some portion of the activities in $\mathcal{A}$ will be allocated to resource type d. We demonstrate this point by setting the social costs such that $c_{a,d} = 0, c_{a,r} = 1$ and $c_{a,m} > c_{a,r}, \forall r \in \mathcal{R}^-$. This means that assigning disadvantaged resources does not consume any social budget, while assigning advantaged humans and computers to tasks will result in non-negative consumption of the social budget (clearly, higher social costs are assigned to computerized tasks). Furthermore, the social budgets are set to satisfy,

$$b_m + \sum_{r \in \mathcal{R}^-} b_r < \sum_{a \in \mathcal{A}} c_{a,m} + \sum_{r \in \mathcal{R}^-} c_{a,r},$$

implying that at least a single activity must be assigned to d. This shows that the COP can be effectively used to guarantee inclusion via training, while limiting automation.

By providing the three *special cases* of setting social policies, we have shown that one can gain insights into the types of choices that the user of our approach must take before applying it in a real-life setting. Furthermore, we demonstrated how the COP representation of SBPR can be useful in supporting decisions in a model-based fashion, thus replacing the need of speculating regarding potential outcomes of setting social policies. Having said the above, configuring parameter values for the underlying redesign problem (setting costs, budgets and weights) is problem-specific. It will also require additional information that must be based on data coming from the application. Therefore, parameter configuration is out of scope for the current work.

6 Conclusion

This work is the first attempt to methodologically include socially responsible practices into the redesign of business processes. Specifically, we provided three main

[5] Counter-intuitively, some automation procedures can be extremely costly, while training disadvantaged employees could be sponsored by the government and thus have negligible costs.

motivations for socially-aware redesign, namely automation, training, and inclusion. To support decision making we formulated social business process redesign (SBPR) as a constrained optimization problem (COP). The resulting COP allows managers to make efficient and socially responsible decisions in complex organisational settings. We have proven that the computational complexity of the SBPR problem is $\mathcal{NP}$-complete and conducted a computational study of our approach to show that it is applicable to large redesign problems. Moreover, we provided a detailed factor analysis of variables that influence the computational complexity of SBPR. Lastly, we demonstrated the impact of organisational policies on the COP and the resulting redesign solutions in real-life scenarios.

Our work paves the way for follow-up research, but has already some practical implications. To start with the latter, we can imagine that our formulation of SBPR can help organizations to make it explicit for themselves what the objective function of their redesign initiative is. We demonstrated that both traditional, efficiency-oriented objectives *and* less traditional objectives, including social or sustainable goals, can be integrated into one and the same redesign initiative. Furthermore, our discussion of the effect of different social policies on SBPR-based decision making (see Sect. 5) shows how tangible support for redesign decisions can be generated. In more concrete terms, we have shown how a range of policies can be mapped onto the characteristics of SBPR to reveal the *nature* of the feasible redesign actions. While the insights that we generated in this way may appear straightforward to the reader, it is our impression that this should be attributed to *hindsight bias* – it is very difficult to foresee these without conducting a quantitative analysis, such as presented in this work.

In future work, we aim at expanding our SBPR model to take into account second-order effects. Specifically, over-qualification may sometimes lead to productivity loss [22], which the current model does not account for. Similarly, the effects of learning decisions may pan out differently for human resources compared to computer resources; productivity growth through AI, for example, may cover multiple years. This would require reformulating SBPR as a dynamic program, taking into account decisions made over a finite time horizon. These decisions must be jointly optimal for every time point, since resource-to-activity and learning decisions at time t influence the corresponding decisions at times $t+1, t+2, \ldots$.

Our model also assumes that an activity may be carried out by a single resource, which is rather strict. It is quite likely that hybrid man-machine teams will become more attractive over time to allocate activities to. The immediate influence of multi-resource activities on our model is at least two-fold. Firstly, we would need to relax the single-activity per resource constraint and consider sets of resources as indices of the decision variable. Secondly, the speedup or slowdown effect of having multiple resources performing a task (compared to a single resource) must be taken into account via the objective function by increasing or decreasing the weights of the allocations. While this may incur more 'unsocial decisions', this development may also benefit individual employees who will be freed of heavy, unhealthy, or dangerous parts of their jobs. This multi-faceted impact also underlines how important it is to advance tools that will help managers to make difficult redesign decisions.

Changing the model to accommodate for additional considerations, such as employee safety and time-dependent redesign, is possible given the expressiveness of constrained optimization problems. However, such changes will require encoding additional features into the model, which may result in a further increase of the computational complexity of SBPR.

References

1. Greening, D.W., Turban, D.B.: Corporate social performance as a competitive advantage in attracting a quality workforce. Bus. Soc. **39**(3), 254–280 (2000)
2. Wang, H., Choi, J.: A new look at the corporate social-financial performance relationship: the moderating roles of temporal and interdomain consistency in corporate social performance. J. Manag. **39**(2), 416–441 (2013)
3. Brieger, S.A., Terjesen, S.A., Hechavarría, D.M., Welzel, C.: Prosociality in business: a human empowerment framework. J. Bus. Ethics **159**(2), 361–380 (2019)
4. Couckuyt, D., Van Looy, A.: Green BPM as a business-oriented discipline: a systematic mapping study and research agenda. Sustainability **11**(15), 4200 (2019)
5. Suša Vugec, D., Tomičić-Pupek, K., Vukšić, V.B.: Social business process management in practice: overcoming the limitations of the traditional business process management. Int. J. Eng. Bus. Manag. **10**, 1847979017750927 (2018)
6. Autor, D.H., Levy, F., Murnane, R.J.: The skill content of recent technological change: an empirical exploration. Q. J. Econ. **118**(4), 1279–1333 (2003)
7. Brynjolfsson, E., McAfee, A.: The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies. WW Norton & Company, New York (2014)
8. Frey, C.B., Osborne, M.A.: The future of employment: how susceptible are jobs to computerisation? Technol. Forecast. Soc. Change **114**, 254–280 (2017)
9. Syed, R., et al.: Robotic process automation: contemporary themes and challenges. Comput. Ind. **115**, 103162 (2020)
10. Becker, G.S.: Human Capital: A Theoretical and Empirical Analysis, with Special Reference to Education, 3rd edn. University of Chicago Press, Chicago (1993)
11. Nedelkoska, L., Quintini, G.: Automation, skills use and training. OECD Social, Employment and Migration Working Papers, No. 202 (2018)
12. Hardy, W., Keister, R., Lewandowski, P.: Technology or upskilling. Trends Task (2016, working paper)
13. Casselman, B., Satariano, A.: Amazon's Latest Experiment: Retraining its Work Force. The New York Times, 11 July 2019 (2019)
14. Illanes, P., Lund, S., Mourshed, M., Rutherford, S., Tyreman, M.: Retraining and reskilling workers in the age of automation. McKinsey Glob. Inst. **29** (2018)
15. Hernandez, B., Keys, C., Balcazar, F.: Employer attitudes toward workers with disabilities and their ADA employment rights: a literature review. J. Rehabil. **66**(4), 4–16 (2000)
16. Graffam, J., Smith, K., Shinkfield, A., Polzin, U.: Employer benefits and costs of employing a person with a disability. J. Vocat. Rehab. **17**(4), 251–263 (2002)
17. Shirani, A.: Upskilling and retraining in data analytics: a skill-adjacency analysis for career paths. Issues Inf. Syst. **20**(4), 65–74 (2019)
18. Fisher, M.L., Jaikumar, R., Van Wassenhove, L.N.: A multiplier adjustment method for the generalized assignment problem. Manag. Sci. **32**(9), 1095–1103 (1986)
19. Plastria, F.: Formulating logical implications in combinatorial optimisation. Eur. J. Oper. Res. **140**(2), 338–353 (2002)

20. Cattrysse, D.G., Salomon, M., Van Wassenhove, L.N.: A set partitioning heuristic for the generalized assignment problem. Eur. J. Oper. Res. **72**(1), 167–174 (1994)
21. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.: MiniZinc: towards a standard CP modelling language. In: Bessière, C. (ed.) CP 2007. LNCS, vol. 4741, pp. 529–543. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74970-7_38
22. Tsang, M.C., Rumberger, R.W., Levin, H.M.: The impact of surplus schooling on worker productivity. Ind. Relat.: J. Econ. Soc. **30**(2), 209–228 (1991)