Optimos: A Tool for Simulation-Driven Business Process Optimization

Orlenys López-Pintado[1], Jannis Rosenbaum[1], Jonas Berx[1] and Marlon Dumas[1]

[1]University of Tartu, Estonia

Abstract

Optimos is a web-based tool that uses optimization techniques to enhance business processes by minimizing cycle times and resource costs through simulation of alternative scenarios. Given a process model in the Business Process Model and Notation (BPMN), a set of simulation parameters, and organizational constraints, Optimos iteratively explores alternative resource allocations and timetables using optimization metaheuristics. The performance of each resource configuration is evaluated via simulation. The target users of Optimos are analysts and managers of business processes. This demo paper describes the scope, architecture, and functionality of Optimos.

Keywords

business process optimization, business process simulation, resource allocation, resource availability timetables

1. Introduction

One primary goal of Business Process Management (BPM) is to improve organizational performance to achieve objectives such as reducing costs and execution times of business processes [1]. For this purpose, what-if analysis can help to identify optimal interventions to maximize or minimize performance measures under specific constraints [2]. For example, an organization can use what-if analysis to explore different staffing schedules within limited working hours, ensuring that resource utilization remains below 80% and employees do not exceed their maximum allowed working hours, balancing labor costs and delivery times.

Business Process (BP) simulation further enhances this capability by providing a powerful technique for assessing what-if scenarios [3]. Using simulation models that act as digital twins, BP simulation captures the organizations' behaviors, offering a detailed representation of real-world processes. These models enable the prediction of the process performance under current configurations (as-is model) or hypothetical changes (to-be model). This capability is essential for optimizing business processes, as it supports process updates that may improve key performance measures, such as resource cost and cycle times [2].

This paper presents Optimos, an optimization tool designed to help process analysts to automatically improve business processes by minimizing cycle times and execution costs through what-if analysis and simulation. Optimos achieves this by adjusting the allocation of resources to activities and modifying resource availability calendars. Additionally, the tool allows analysts to define organizational constraints that restrict the optimization of their resource allocations.

Achieving an ideal resource configuration that minimizes costs and cycle times is challenging. Typically, no single configuration can improve both objectives simultaneously. For instance, increasing the number of resources reduces activity waiting times, shortening the overall cycle time. Similarly, if resources are less busy due to higher availability or extended working hours, waiting times decrease, making processes faster. However, having more resources and increased working hours also increases labor costs. On the other hand, reducing the number of resources or limiting their working hours

Proceedings of the Best BPM Dissertation Award, Doctoral Consortium, and Demonstrations & Resources Forum co-located with 22nd International Conference on Business Process Management (BPM 2024), Krakow, Poland, September 1st to 6th, 2024.

*Corresponding authors.

orlenys.lopez.pintado@ut.ee (O. López-Pintado); jannis.rosenbaum@ut.ee (J. Rosenbaum); marlon.dumas@ut.ee (M. Dumas)

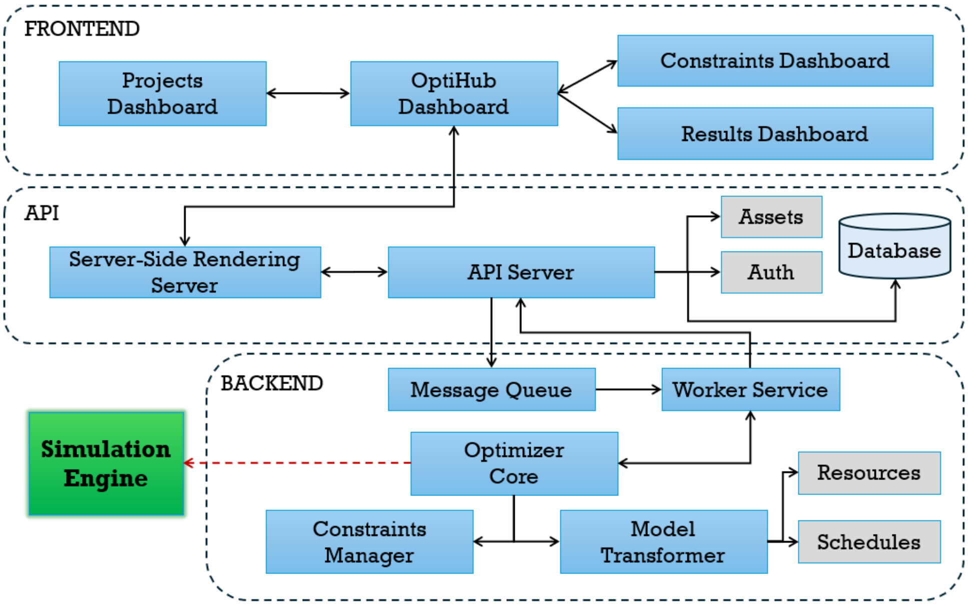


Figure 1: Optimos Architecture.

lowers costs but increases waiting times due to busier remaining resources. This trade-off between cost and time is a key consideration in optimizing business processes [3].

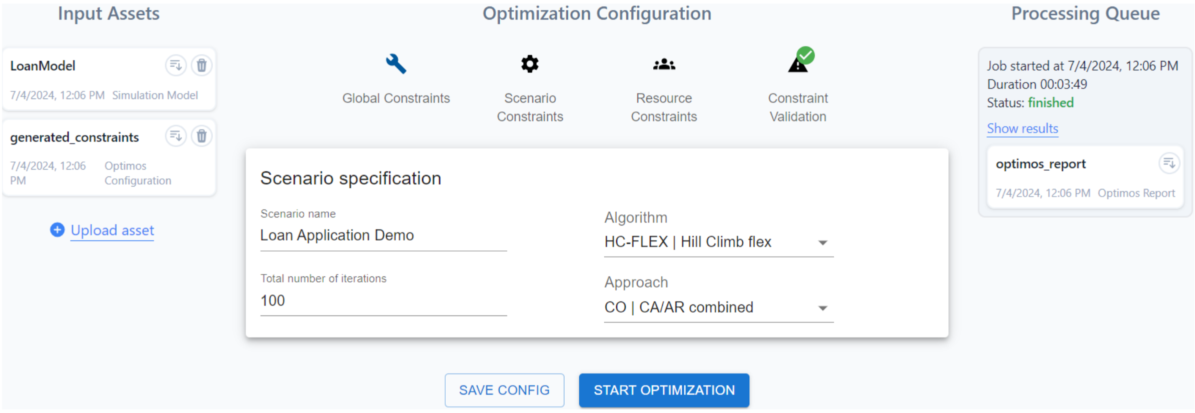
Manually balancing the number of resources and their working hours is challenging for analysts because it involves considering multiple combinations of variables and constraints. The technical complexity of exploring a wide range of potential solutions makes finding optimal resource allocations difficult without automated tools. Optimos aids analysts by automating the process, allowing them to optimize their resources without requiring any technical background.

2. Optimos Architecture

Optimos is a Web-based tool logically structured into three layers, as shown in Figure 1. The FRONTEND layer includes several dashboards that enable user interaction. The `Projects Dashboard` is the entry point for users to manage different optimization projects. Within each project, users navigate to the `OptiHub Dashboard`, the central hub, displaying the optimization status and current configurations and allowing users to start new optimizations. From the `OptiHub Dashboard`, users can access the `Constraints Dashboard` to define and update optimization constraints or the `Results Dashboard` to view suggested resource allocations and statistics from ongoing or completed optimizations.

The API layer is the intermediary between the FRONTEND and BACKEND layers. The `Server-Side Rendering Server` retrieves fully-rendered HTML pa-ges to improve performance. The `API Server` manages business logic, data processing, and secure interactions with the BACKEND. This separation allows each server to be scaled and maintained independently. The `API Server` also handles `Assets` provided by the users, such as process models, simulation parameters, or optimization results. Additionally, it manages authentication (`Auth`) to ensure secure access to the system, verifying user credentials and maintaining session security. Finally, the `API Server` interacts with a `Database`, storing and retrieving essential data such as user information, project details, constraints, and results to ensure data persistence in the system.

The BACKEND layer is where the core optimization processes occur. The `Message Queue` manages communication and task distribution between the `API server` and BACKEND services, ensuring smooth processing of tasks. The `Worker Service` executes the tasks assigned by the `Message Queue`. The `Optimizer Core` is the central component responsible for performing the optimization calculations and assembling and analyzing the results delivered by each other component. It works closely with the



LoanModel
7/4/2024, 12:06 PM Simulation Model

generated_constraints
7/4/2024, 12:06 PM Optimos Configuration

Scenario specification

Scenario name: Loan Application Demo

Total number of iterations: 100

Algorithm: HC-FLEX | Hill Climb flex

Approach: CO | CA/AR combined

Job started at 7/4/2024, 12:06 PM
Duration 00:03:49
Status: finished

optimos_report
7/4/2024, 12:06 PM Optimos Report

Figure 2: Partial view of OptiHub Dashboard.

`Constraints Manager`, which manages user-defined constraints, ensuring that optimizations respect these rules. The `Model Transformer` adjusts (if possible) models to fix potentially problematic elements, like bottleneck activities with high waiting times or over- and underutilized resources, aiming to find an improved model. The `Simulation Engine` performs simulations of each transformed model. It retrieves the results to the `Optimizer Core`, which assesses the impact of different resource allocations and schedules and accordingly updates the set of optimal solutions.

3. Functionality

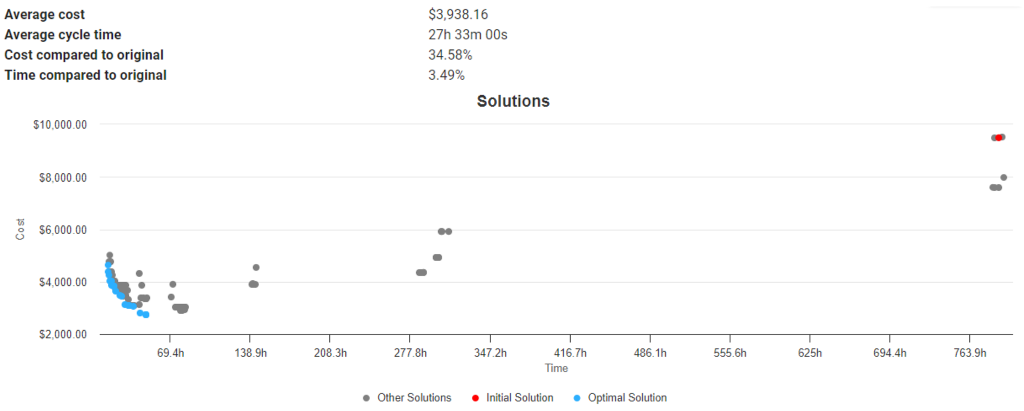
To interact with Optimos, users must authenticate to access their `Projects Dashboard`. From there, they can view or create optimization projects. By clicking on a specific project, users are redirected to the `OptiHub Dashboard`, where they can edit or delete the selected project.

An optimization project consists of one or multiple simulation models, each with a BPMN model [4] representing the process control flow and a JSON file with the simulation parameters according to [3]. The simulation models supported by Optimos can be generated by Simod following the instructions from the code repository [1] and uploaded to the project via `Upload asset`. By clicking the desired simulation model in the `OptiHub Dashboard`, users can define constraints to restrict the optimization process. As shown in Figure 2, constraints are divided into `Global`, `Scenario`, and `Resource Constraints`.

The `Global Constraints` (the tab shown in Figure 2) relate to the optimization settings. They include the maximum number of iterations, the choice of the search algorithm, and the optimization approach to follow. An iteration refers to one cycle of evaluating potential solutions. Depending on the selected “Algorithm”, Optimos keeps only optimal solutions at the current iteration and discard the rest (option HC-STRICT), retains near-optimal solutions for future iterations (HC-FLEX), or considers all solutions until the iteration limit is reached (TS). The selected “Approach” specifies whether to only add/remove resources (AR), only update schedules (CA), or combine both. Optimos allows three ways of combining them: (1) try to add/remove resources and update schedules simultaneously at each iteration (CO), (2) first add/remove resources and then improve schedules (ARCA), or (3) first optimize schedules and then adjust the number of resources (CAAR). This allows the optimization process to focus strictly on improvements or explore a broader range of potential solutions.

Under the `Scenario Constraints` tab, users can set general restrictions for all resources, such as maximum weekly hours, consecutive working hours (shifts), and daily working limits.,e.g., to define a rule that resources can work at most in two shifts of 4 hours on working days, with some rest in between. Then, the `Resource Constraints` tab defines conditions specific to each resource, such as personal capacity limits and maximum shifts they can work daily or weekly. Additionally, it allows users to restrict intervals of a day in which a given resource must always work or, conversely, never work. For example, users can specify that a particular part-time resource can work at most five shifts of 4 hours per week and must always be available from 9 AM to 12 PM from Monday to Friday but

[1] https://github.com/AutomatedProcessImprovement/Simod



Average cost $3,938.16
Average cycle time 27h 33m 00s
Cost compared to original 34.58%
Time compared to original 3.49%

Figure 3: General optimization results: aggregating costs and cycle times from all the solutions.

never work after 1 PM or on weekends. Finally, the `Constraint Validation` tab displays and allows to correct all the potential errors on the constraint definitions, e.g., if they contradict the current calendars of a given resource in the simulation model. If all the constraints are valid (green), users can start the optimization by clicking the corresponding button.

Users can view the processing queue for ongoing and completed simulations on the right side of the `OptiHub Dashboard` (see Figure 2). By clicking on an entry, they can access detailed optimization reports. These reports show metrics like estimated mean costs, mean times, and average resource utilization achieved at each iteration, helping users compare improvements to the original process.

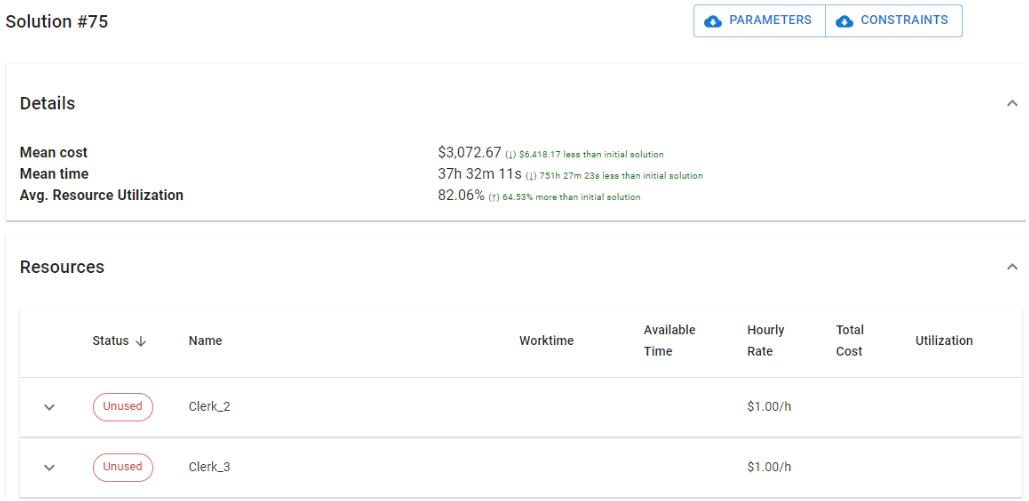
For example, Figure 3 illustrates the general results after optimizing a business process, including a scatter plot, average costs and cycle times, and comparisons to the initial solution. The scatter plot shows the performance achieved by the solutions explored by the optimizer, with the y-axis representing cost and the x-axis representing cycle time. It highlights the initial solution (red dot), optimal solutions (blue dots), and other (sub-optimal) solutions (gray dots). This visual plot aids users in getting general insights into the distribution of costs and cycle times and how they improve across iterations. Also, by clicking any solution in the chart, optimal or not, users can get more detailed information for each resource allocation, such as assigned tasks, total working times, hourly rates, and timetable changes.

Next, Figure 4 displays the information regarding a specific solution, i.e., corresponding to the 75th allocation assessed by the optimizer. While the values in the general chart (Figure 3) aggregate all the optimal allocations, the values shown in Figure 4 focus only on the costs and cycle times achieved with the current allocation. It also shows average resource utilization and improvements over the original allocation. Specifically, in this allocation, the optimizer detected that two clerks are unused, suggesting they are unnecessary, and their exclusion leads to improvements. Indeed, the unused clerks' dynamic information, like utilization, cost, and available time, is missing as they did not participate in the simulated execution. In the example used for this demo, four clerks were underutilized initially, while only one credit officer was overutilized, causing increased waiting times and a bottleneck. Therefore, although not shown in the figure, the optimizer also identified that adding more credit officers or extending their schedules could lead to an optimal allocation. Users can also click on any resource (e.g., Clerk-2) to see allocated tasks and their adjusted timetables after optimization.

In this way, analysts can understand how each iteration adjusts the resource allocations and timetables to enhance performance. Users can review these results and choose a resource allocation from the solutions produced at each iteration based on their preferences for balancing time and cost. Optimos allows users to stop the optimization process at any point if they are satisfied with the results.

4. Availability and Maturity

Optimos is accessible as a cloud service from https://pix-w1.cloud.ut.ee/. Documentation of Optimos and the source code of the Backend can be found at https://github.com/AutomatedProcessImprovement/



Solution #75

Details

Mean cost	$3,072.67 (↓) $6,418.17 less than initial solution
Mean time	37h 32m 11s (↓) 751h 27m 23s less than initial solution
Avg. Resource Utilization	82.06% (↑) 64.53% more than initial solution

Resources

	Status ↓	Name	Worktime	Available Time	Hourly Rate	Total Cost	Utilization
⌄	Unused	Clerk_2			$1.00/h		
⌄	Unused	Clerk_3			$1.00/h		

Figure 4: Information of a specific solution (resource allocation) retrieved by the optimizer.

roptimus-prime. This GitHub repository contains links to separate repositories for the frontend and backend (API) layers. A demonstration video of Optimos is available at https://youtu.be/jBtBEwO-wT4. The example file containing the simulation model and constraints used in the video can be found in the Optimos repository: https://github.com/AutomatedProcessImprovement/roptimus-prime/raw/main/test_assets/demo/Demo%20Files.zip.

Optimos implements the optimization approaches presented in [5, 3]. The first paper [5] describes a basic model with resources grouped in pools. The optimizer may add or remove resources to reduce cycle times and costs. The second paper [3] proposes a more sophisticated model with differentiated resources, where each resource has its own availability and performance. The optimizer may add or remove resources or change the timetable of a resource while ensuring that the optimized timetables fulfill a given set of constraints (e.g., no more than 40 hours per week). Optimos relies on Prosimos [6, 3] as a simulation engine to estimate the impact after transforming the model in each iteration.

Acknowledgments

Work funded by European Research Council (PIX project) and the Estonian Centre of Excellence in Artificial Intelligence funded by the Estonian Ministry of Education and Research.

References

[1] M. Dumas, M. L. Rosa, J. Mendling, H. A. Reijers, Fundamentals of Business Process Management, Second Edition, Springer, 2018.

[2] M. Dumas, Constructing digital twins for accurate and reliable what-if business process analysis, in: BPM Workshops, volume 2938 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 23–27.

[3] O. López-Pintado, M. Dumas, J. Berx, Discovery, simulation, and optimization of business processes with differentiated resources, Inf. Syst. 120 (2024) 102289.

[4] O. M. Group, Business Process Model and Notation (BPMN), Version 2.0.2, 2013. URL: http://www.omg.org/spec/BPMN/2.0.2/.

[5] O. López-Pintado, M. Dumas, M. Yerokhin, F. M. Maggi, Silhouetting the cost-time front: Multi-objective resource optimization in business processes, in: BPM Forum, volume 427 of *LNBIP*, Springer, 2021, pp. 92–108.

[6] O. López-Pintado, I. Halenok, M. Dumas, Prosimos: Discovering and simulating business processes with differentiated resources, in: EDOC Demos, volume 466 of *LNBIP*, Springer, 2022, pp. 346–352.