Generating Multi-objective Optimized Business Process Enactment Plans

Andrés Jiménez-Ramírez[1], Irene Barba[1],
Carmelo del Valle[1], and Barbara Weber[2]

[1] University of Seville, Dpto. Lenguajes y Sistemas Informáticos, Spain
{ajramirez,irenebr,carmelo}@us.es
[2] University of Innsbruck, Department of Computer Science, Austria
barbara.weber@uibk.ac.at

Abstract. Declarative business process (BP) models are increasingly used allowing their users to specify what has to be done instead of how. Due to their flexible nature, there are several enactment plans related to a specific declarative model, each one presenting specific values for different objective functions, e.g., completion time or profit. In this work, a method for generating optimized BP enactment plans from declarative specifications is proposed to optimize the performance of a process considering multiple objectives. The plans can be used for different purposes, e.g., providing recommendations. The proposed approach is validated through an empirical evaluation based on a real-world case study.

Keywords: Business Process Management, Constraint Programming, Planning and Scheduling, Constraint-based BP Models.

1 Introduction

Nowadays, there exists a growing interest in aligning information systems in a process-oriented way [28] as well as in the effective and flexible management of business processes (BPs) [22]. A BP consists of a set of activities which are performed in coordination in an organizational and technical environment [28], and which jointly realize a business goal. Typically, BPs are specified in an imperative way. However, declarative BP models (e.g., constraint-based models) are increasingly used allowing their users to specify what has to be done instead of how [19]. Declarative specifications facilitate the human work involved, avoid failures, and allow to obtain a better optimization, since the tacit nature of human knowledge is often an obstacle to eliciting accurate BP models [10].

Due to their flexible nature, frequently several ways to execute declarative process models exist, i.e., there are several enactment plans related to a specific declarative model, each one presenting specific values for different objective functions, e.g., overall completion time or profit. The decision about the way to execute this model can be quite challenging since usually many constraints need to be obeyed, multiple instances of a process get concurrently executed within a particular timeframe, shared resources need to be allocated, and relevant objective functions should be considered.

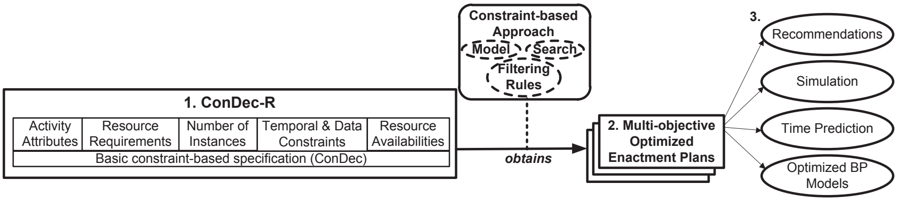


Fig. 1. Overview of our Approach

In this work, a method for generating optimized BP enactment plans from declarative specifications is proposed to optimize the performance of a process by considering multiple objectives. For this, we propose an extension of ConDec [19] motivated by requirements described in literature [18,29] (i.e., dealing with temporal and data constraints) and imposed by the case studies we have conducted (i.e., dealing with activity attributes, resource management), named ConDec-R. Specifically, we extend ConDec by considering (1) temporal and data constraints, (2) resource requirements for activity executions, and (3) estimates for some activity attributes (e.g., duration), number of process instances executed within a particular timeframe, and resource availabilities.

Figure 1 provides an overview of our approach. Taking the ConDec-R specification as a starting point (cf. Fig. 1(1)), multi-objective optimized enactment plans can automatically be generated (cf. Fig. 1(2)). For this, activities to be executed have to be selected and ordered (planning problem [12]) considering all the constraints, resource requirements, and estimates regarding the number of instances executed within a particular timeframe, resource availabilities, and some activity attributes, e.g., activity durations (scheduling problem [5]). For planning and scheduling (P&S) the activities such that the process objective functions are optimized, a constraint-based approach is proposed. The generated plans can leverage the BP management (BPM) life cycle [28], since they can be used for different purposes (cf. Fig. 1(3)), e.g., recommendations [3], simulation [24], time prediction [27], and generation of optimized BP models [21].

The main contributions of this paper are: (1) the definition of a new language for the constraint-based specification of BPs (cf. Sect. 3, Fig. 1(1)), (2) the automatic generation of multi-objective optimized BP enactment plans from ConDec-R specifications through a constraint-based approach (cf. Sect. 4, Fig. 1(2)), (3) the application of the proposed approach to a case study (cf. Sect. 5), and (4) its empirical evaluation (cf. Sect. 6).

Initial aspects of the proposed approach have been previously presented (cf. [14]). However, this paper significantly extends [14] by: (1) providing improved expressiveness through choice templates [19], metric temporal constraints [18,29], and data constraints [18], (2) extending the constraint-based approach by including new filtering rules (cf. Sect. 2) for the aforementioned constraints, (3) dealing with alternative resources to enable activities to be performed by different resources, (4) considering multiple objective functions in the optimization, and (5) evaluating the proposal in the context of an actual process, and therefore demonstrating that it can work in practice for managing realistic problems.

Section 2 introduces backgrounds, Sect. 3 details the ConDec-R language, Sect. 4 shows the generation of optimized plans, Sect. 5 explains a real example, Sect. 6 deals with the evaluation, Sect. 7 presents a critical discussion, Sect. 8 summarizes related work, and Sect. 9 includes some conclusions and future work.

2 Background

Different paradigms for process modelling exist, e.g., imperative and declarative. Imperative process models are well structured representations which specify exactly *how* things have to be done by explicitly depicting all possible behavior. A declarative model, in turn, is a loosely-structured representation focused on *what* should be done which specifies all forbidden behavior. Therefore, declarative models are commonly used for representing processes with high variability which can be executed in several ways. We use the declarative language ConDec [19] for specifying constraint-based models (cf. Def. 1) since it allows the specification of activities together with the constraints which must be satisfied for correct BP enactment and for the goal to be achieved. Moreover, ConDec allows to specify a wide set of BP models in a simple and flexible way. Constraints can be added to a ConDec model to specify forbidden behavior, restricting the desired behavior (cf. [19]). ConDec templates are grouped into:

1. **Existence** constraints: unary relations concerning the number of times one activity is executed, e.g., Exactly(N,A) specifies that A must be executed exactly N times.
2. **Relation** constraints: positive binary relations used to impose the presence of a certain activity when some other activity is performed, e.g., Precedence(A,B) specifies that to execute activity B, activity A needs to be executed before.
3. **Negation** constraints: negative relations used to forbid the execution of activities in specific situations, e.g., NotCoexistence(A,B) specifies that if B is executed, then A cannot be executed, and vice versa.
4. **Choice** constraints: n-ary constraints expressing the need of executing activities belonging to a set of possible choices, e.g., ExactlyChoice(N,{A,B,C}) specifies that exactly N activities of the set {A,B,C} must be executed.

Definition 1. *A* ***constraint-based process model*** $S = (A, C_{BP})$ *consists of a set of activities* A*, and a set of executing behavior constraints* C_{BP}*. Each activity* $a \in A$ *are executed arbitrarily often if not restricted by any constraint.*

To support increased expressiveness of ConDec, several proposals for extensions have been made like metric temporal constraints [18,29] or data relations [18], which are all supported by our proposal. As an example, the temporal constraint Precedence(A,B,[5,10]) specifies that to start the execution of B, A needs to be finished between 5 and 10 time units before. Using data constraints, for example, the earliest and the latest start and end times of an activity, together with the selections of the choice template can be constrained through input data. As an example, the data constraint A.startTime≥Data.T specifies that A can only start after time T of input Data (for more examples see Sect. 5).

Due to their flexible nature, there are frequently different ways to execute a constraint-based model in such a way that all constraints are fulfilled. The different valid execution alternatives, however, can vary significantly in how well different performance objective functions (cf. Def. 2) can be achieved.

Definition 2. *An* ***objective function*** *OF of a BP is a function to be optimized during the BP enactment, e.g., maximization of the profit.*

For generating plans optimizing the objective functions of constraint-based process models, activities to be executed have to be planned [12] and scheduled [5]. To do this, a constraint-based approach is proposed (cf. Sect. 4).

The area of scheduling [5] includes problems in which it is necessary to determine an enactment plan for a set of activities related by constraints (in our context the control-flow constraints, together with the resource, data and temporal constraints). Several objective functions are usually considered to be optimized, e.g., minimization of the overal completion time. In a wider perspective, in planning [12], the activities to be executed are not established a priori, hence it is necessary to select them from a set of alternatives and to establish an ordering.

Constraint Programming (CP) [23] supplies a suitable framework for modelling and solving P&S problems [26]. To solve a problem through CP, it needs to be modelled as a constraint satisfaction problem (CSP, cf. Def. 3).

Definition 3. *A* ***CSP*** *$P = (V, D, C_{CSP})$ is composed of a set of variables V, a domain of values D for each variable in V, and a set of constraints C_{CSP} between variables, so that each constraint represents a relation between a subset of variables and specifies the allowed combinations of values for these variables.*

To improve the modelling of the problems global constraints, i.e., constraints capturing a relation between a non-fixed number of variables, can be defined.

A solution to a CSP consists of assigning values to CSP variables, being feasible when the assignments satisfy all the constraints.

Since actual problems typically involve multiple conflicting objective functions (cf. Def. 2), multi-objective constraint optimization problems (MO-COPs, cf Def. 4) are considered in the current work. In MO-COPs, usually no unique optimal solution exists, but a set of Pareto optimal solutions (cf. Def. 5) can be found.

Definition 4. *A* ***MO-COP*** *$P_o = (V, D, C_{CSP}, OFs)$ is a CSP which also includes a set of objective functions OFs to be optimized.*

Definition 5. *A solution for a MO-COP is* ***Pareto optimal*** *when it is not dominated by any other solution, i.e., for obtaining a better feasible solution in one of the objective functions, at least another objective needs to be deteriorated.*

To solve multi-objective optimization problems (for more information, the reader is referred to [9]), there are, basically, three approaches: (i) defining a new objective function (i.e., combining the original objective functions) which can be optimized with single objective solvers (e.g., the weighted-sum method [1]), (ii) optimizing one of the objective functions constraining the other ones (e.g., ε-constraint method [13]), and (iii) working with a set of Pareto optimal solutions (e.g., evolutionary multi-objective optimization [6]). In this work, the ε-constraint method [13] is applied since it appeared well suited for our purposes and typically provides good results.

Regardless of the used search method, the global constraints can be implemented through filtering rules (i.e., rules responsible for removing values which do not belong to any solution) to efficiently handle the constraints in the search for solutions.

3 ConDec-R: A Constraint-Based BP Language

To specify the processes in a declarative way, ConDec [19] is used as basis (cf. Sect. 2). Motivated by requirements described in literature [18,29] as well as the necessities of the case studies we have conducted (cf. Sect. 5) we extend ConDec to ConDec-R. Besides extending ConDec with resource reasoning and estimates for activity durations (which are partially covered in previous works [2,14]), ConDec-R supports activities with an open set of attributes and alternative resources (cf. Def. 6), and choice, temporal and data constraints. In a ConDec-R process model (cf. Def. 7), all the previously stated extensions are considered.

Definition 6. *A **BP activity** $BPAct = (a, Res, Atts)$ represents a BP activity called a, which can be performed by any resource included in Res*[1]*, and which has a set of attributes associated (e.g., duration and profit) which is composed of tuples <att, val> (i.e., Atts).*

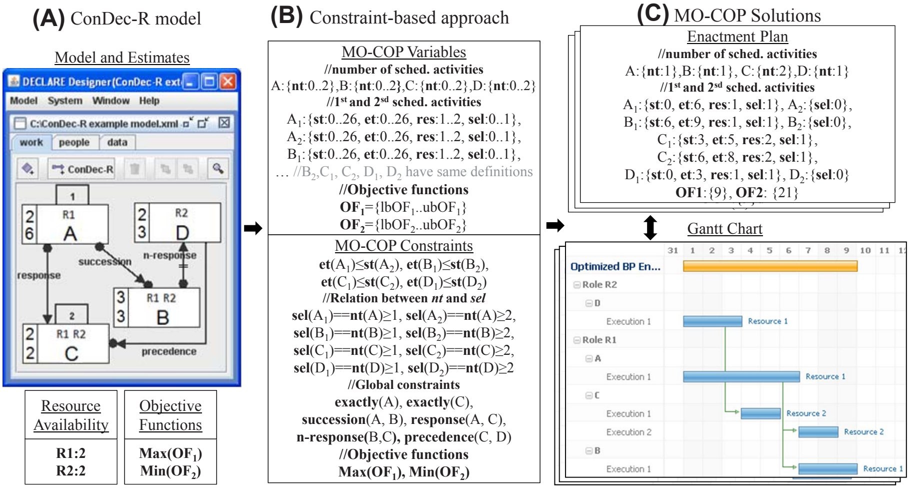
Definition 7. *A **ConDec-R process model** CR = (BPActs, Data, C_{BP}, Av-Res, OFs) related to a constraint-based process model S = (A,C_{BP}) (cf. Def. 1) is composed of (1)a set of BP activities (cf. Def. 6) BPActs, (2) problem data information Data, (3) a set of ConDec constraints C_{BP} which relates activities included in BPActs and/or the data included in Data, (4) a set of available resources Res which is composed of tuples (role,#role) which includes for each role (i.e., role) the number #role of available resources*[2]*, and (5) a set of the objective functions OFs to be optimized (cf. Def. 2).*

Figure 2(A) shows a simple ConDec-R model[3] (cf. Def. 7) where: $BPActs = \{(A, < R1 >, << att_1, 2 >, < att_2, 6 >>), (B, < R2 >, << att_1, 2 >, < att_2, 2 >>), (C, < R1, R2 >, << att_1, 2 >, < att_2, 3 >>), (D, < R1, R2 >, << att_1, 3 >, < att_2, 2 >>)\}$; $Data = \{\}$; $C_{BP} = \{$ $exactly(1, A)$, $exactly(2, B)$, $succession(A, B)$, $response(A, B)$, *negate-response(B,C)*, $precedence(C, D)$ $\}$; $Res = \{(R1, 2), (R2, 2)\}$; and $OFs = \{OF_1, OF_2\}$.

[1] This allows activities to be performed by alternative resources, whereas in previous works (cf. [2,14]) only one resource can be assigned to each activity.

[2] The role-based allocation pattern [25] is considered.

[3] We extend Declare [7] (i.e, a workflow management system that can be used to specify ConDec models) to allow specifying ConDec-R models.



Resource Availability	Objective Functions

R1:2
R2:2
	Max(OF$_1$)
	Min(OF$_2$)

MO-COP Variables

//number of sched. activities

A:{**nt**:0..2},B:{**nt**:0..2},C:{**nt**:0..2},D:{**nt**:0..2}

//1st and 2sd sched. activities

A_1:{**st**:0..26, **et**:0..26, **res**:1..2, **sel**:0..1},
A_2:{**st**:0..26, **et**:0..26, **res**:1..2, **sel**:0..1},
B_1:{**st**:0..26, **et**:0..26, **res**:1..2, **sel**:0..1},
… //B_2,C_1, C_2, D_1, D_2 have same definitions

//Objective functions

OF$_1$={lbOF$_1$..ubOF$_1$}
OF$_2$={lbOF$_2$..ubOF$_2$}

MO-COP Constraints

et(A_1)≤**st**(A_2), **et**(B_1)≤**st**(B_2),
et(C_1)≤**st**(C_2), **et**(D_1)≤**st**(D_2)

//Relation between *nt* and *sel*

sel(A_1)==**nt**(A)≥1, **sel**(A_2)==**nt**(A)≥2,
sel(B_1)==**nt**(B)≥1, **sel**(B_2)==**nt**(B)≥2,
sel(C_1)==**nt**(C)≥1, **sel**(C_2)==**nt**(C)≥2,
sel(D_1)==**nt**(D)≥1, **sel**(D_2)==**nt**(D)≥2

//Global constraints

exactly(A), **exactly**(C),
succession(A, B), **response**(A, C),
n-response(B,C), **precedence**(C, D)

//Objective functions

Max(OF$_1$), Min(OF$_2$)

Enactment Plan

//number of sched. activities

A:{**nt**:1},B:{**nt**:1}, C:{**nt**:2},D:{**nt**:1}

//1st and 2sd sched. activities

A_1:{**st**:0, **et**:6, **res**:1, **sel**:1}, A_2:{**sel**:0},
B_1:{**st**:6, **et**:9, **res**:1, **sel**:1}, B_2:{**sel**:0},
C_1:{**st**:3, **et**:5, **res**:2, **sel**:1},
C_2:{**st**:6, **et**:8, **res**:2, **sel**:1},
D_1:{**st**:0, **et**:3, **res**:1, **sel**:1}, D_2:{**sel**:0}
OF1:{9}, **OF2**: {21}

Fig. 2. Generating Optimized Enactment Plans from ConDec-R Models

4 From ConDec-R to Optimized Enactment Plans

To generate optimal (or optimized) execution plans for a specific ConDec-R model, we propose a constraint-based approach for P&S the BP activities. This includes: the modelling of the problem as a MO-COP, the use of global constraints implemented through filtering rules to improve the modelling of the problems and to efficiently handle the constraints in the search for solutions, and a search algorithm for solving the MO-COP.

Representing the ConDec-R model as a MO-COP: Given a process modeled as a ConDec-R model (cf. Def. 7, Fig. 2(A)), it needs to be represented as a MO-COP (cf. Def. 4, Fig. 2(B)). Regarding the proposed MO-COP model, BP activities (repeated activities in the MO-COP model, cf. Def. 8), which can be executed arbitrarily often if not restricted by any constraint, are modelled as a sequence of optional scheduling activities (cf. Def. 9). This is required since each execution of a BP activity (i.e., a scheduling activity) is considered as one single activity which needs to be allocated to a specific resource and temporarily placed in the enactment plan, i.e., stating values for its start and end times.

Definition 8. *A* ***repeated activity*** $ra = (a, Res, Att, nt)$ *is a BP activity* $BPAct = (a, Res, Atts)$ *(cf. Def. 6) which can be executed several times. It defines a CSP variable which specifies the number of times the BP activity is executed (i.e.,* nt*).*

Definition 9. *A* ***scheduling activity*** $sa = (st, et, res, sel)$ *related to a repeated activity* $ra = (a, Res, Att, nt)$*, represents a specific execution of* ra*, where* st *and* et *are CSP variables indicating the start and the end times of the activity execution, respectively,* $res \in Res$ *is a CSP variable representing the resource* *used for the execution, and sel is a CSP variable indicating whether or not the activity is selected to be executed (i.e., equal to 0 in the case that it is not executed and equal to 1 otherwise).*

For each repeated activity, nt_{MAX}[4] scheduling activities exist, which are added to the CSP problem specification, apart from including a variable nt.

Moreover, additional CSP variables representing the objective functions to optimize are also included in the MO-COP (cf. Fig. 2(B)). In this way, the ConDec-R model $CR = (BPActs, Data, C_{BP}, AvRes, OFs)$ (cf. Def. 7) is transformed into a MO-COP P_o = *(V, D, C_{CSP}, OFs)* (cf. Def. 4, Fig. 2(B)) where:

1. $V = \{nt(a), a \in BPActs\} \cup \{st(a_i), et(a_i), res(a_i), sel(a_i), i \in [1..\, nt_{MAX}(a)], a \in BPActs\} \cup OFs$.
2. D is composed of the domains of each CSP variable var, where $UB(var)$ and $LB(var)$ represent the upper and lower bounds of the domain of var, respectively. In the example of Fig. 2, the domain [0..2] is used for nt since 2 is the maximum cardinality for the BP activities (established by existence relations in the constraint-based model). The domain [0..26] is used for et and st since 26 would be the completion time if all the scheduling activities were serially executed taking the maximum cardinality for the BP activities into account.
3. C_{CSP} is composed of the resource constraints, the global constraints (implemented by the filtering rules, cf. Sect. 2) related to C_{BP}, and the constraints which are inherent to the proposed model:
   (a) $\forall a \in BPActs\ \forall i : 1 \leq i < nt(a) : et(a_i) \leq st(a_{i+1})$ (i.e., a specific execution of a repeated activity precedes the next execution of the same activity).
   (b) $\forall a \in BPActs\ \forall i : 1 \leq i \leq UB(nt(a)) : sel(a_i) == nt(a) >= i$ (i.e., the nt variable of the repeated activity is directly related to the sel variables of the associated scheduling activities).

Resource constraints are not explicitly stated since most constraint-based systems provide a high-level constraint modeling specific to scheduling which includes an efficient management of shared resources. Besides the role-based allocation pattern, the CSP variables which are included in the model can be also used for specifying further resource constraints [25].

Filtering Rules: Many constraint-based approaches for modelling and solving P&S problems have been proposed [23]. Moreover, several proposals exist for filtering rules related to specialized scheduling constraints (e.g., [16,4]). Therefore, the considered problem could be managed by adapting existing constraint-based approaches. However, some ConDec-R templates entail complex reasoning about several combined innovative aspects, such as the alternating executions of activities together with the varying number of times which these activities are executed. Therefore, we implemented our own specific global constraints through innovative filtering rules to facilitate the specification of the problems and to

[4] nt_{MAX} represents the maximum value for the initial domain of nt (cf. Fig 2(B)).

```
TemporalPrecedence(A,B,[min, max]) ->
    If LB(nt(B)) > 0 then
        nt(A) <- nt(A) - {0}
    If LB(et(act(A,1))) + min > LB(st(act(B,1))))then
        LB(st(act(B,1)))) <- LB(et(act(A,1))) + min
    If UB(et(act(A,1))) - max > UB(st(act(B,1))) then
        UB(et(act(A,1))) <- UB(st(act(B,1))) - max
```

Fig. 3. Propagator for Temporal Precedence Template in ConDec-R

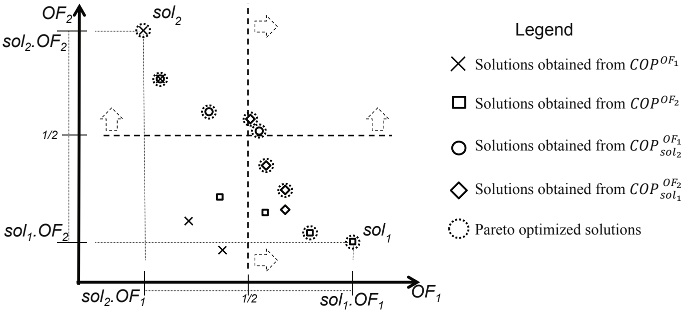


Fig. 4. Example of a MO-COP with two objective functions which is solved through four single objective COPs

increase the efficiency in the search for solutions. In this way, the constraints stated in the ConDec-R specification (cf. Def. 7) are included in the MO-COP model through the related global constraints (cf. Fig. 2(B)). In the MO-COP, initial estimates are made for upper and lower bounds of variable domains, and these values are refined during the search process by the developed filtering rules.

In this work, filtering rules related to constraints which were not considered in previous works[5] (i.e., [2,14]) have been developed, i.e., choice, temporal and data. As an example, the $TemporalPrecedence(A, B, [min, max])$ rule is shown in Fig. 3, where the propagator that describes the pruning of domains appears after symbol →. This constraint means that between min and max units of time before the first execution of B, at least one execution of A must be executed.

Solving the MO-COP: Once the problem is modelled, several constraint-based mechanisms can be used to obtain the solutions for the MO-COP (cf. Def. 4), i.e., multi-objective optimized enactment plans (cf. Def. 10, Fig. 2(C)). As stated in Sect. 2, we implemented a multi-objective optimization search algorithm based on the ε-constraint method [13].

In Fig. 4 the search algorithm is applied over a problem with two objective functions (OF_1 and OF_2). For this example, our approach performs the following steps:

[5] A detailed description of the developed basic ConDec-R filtering rules can be found at `http://regula.lsi.us.es/MOPlanner/FilteringRules.pdf`

1. Two single-objective constraint optimization problems (COPs) are generated (i.e., COP^{OF_1} and COP^{OF_2}) with the goal of optimizing (maximizing in this example) OF_1 and OF_2 respectively.
2. These two COPs are solved and therefore an optimized solution is found for each one (i.e., sol_1 and sol_2 respectively) along with a set of intermediate solutions.
3. Two new COPs (i.e., $COP^{OF_2}_{sol_1}$ and $COP^{OF_1}_{sol_2}$) which include a constraint over the value of the objective function that is not optimized (i.e., $OF_2 \geq |sol_1.OF_2 - sol_2.OF_2|/2$ and $OF_1 \geq |sol_2.OF_1 - sol_1.OF_1|/2$ respectively[6]) are generated.
4. These two new COPs are solved and new solutions are obtained.
5. From all the obtained solutions, those solutions which are not Pareto optimal (cf. Def. 5) are removed.

Definition 10. *A **BP enactment plan** consist of: (i) the number of times each repeated activity is executed, (ii) the start and end times for each activity execution (i.e., scheduling activity), and (iii) the resource which is used for each scheduling activity.*

The generated enactment plans can be graphically represented by a Gantt chart (cf. Fig. 2(C)). This chart illustrates the activity schedules and allows users to understand the solution at a glance. Moreover, the relations between executions of activities are depicted in the Gantt chart due to the ConDec-R constraints of the model (e.g., the relation between the first execution of D and the first execution of C is due to the constraint $precedence(C, D)$).

Since the generation of optimized plans presents NP-complexity [11], it is not possible to ensure the optimality of the generated plans for all the cases. The developed constraint-based approach, however, allows solving the considered problems in an efficient way (cf. Sect. 6).

5 A Real Example: A Beauty Salon of Seville

Motivation: The considered business (i.e., a beauty salon) has grown considerably in the last years. It has expanded from a small salon with three employees to six and included additional facilities to be able to offer additional services. These changes, including the quick growth together with the complex constraints which need to be obeyed, resulted in problems related to the management of the salon. In particular, long waiting time for clients and missing schedules for employees are causing problems, affecting customer satisfaction and profit of the business.

Improving the Management: The goal of the business is to improve the current situation by using a constraint-based approach for optimizing its BPs. Since our approach generates optimized enactment plans, optimized schedules for employees can be suggested, and therefore, the aforementioned problems can

[6] The symbol $\geq$ is used since maximization of the objective functions is considered in the example of Fig. 4. The symbol $\leq$ would be used for minimization.

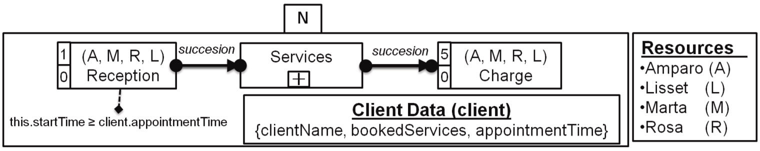


Fig. 5. ConDec-R Model for the Beauty Salon Problem (Top level process)

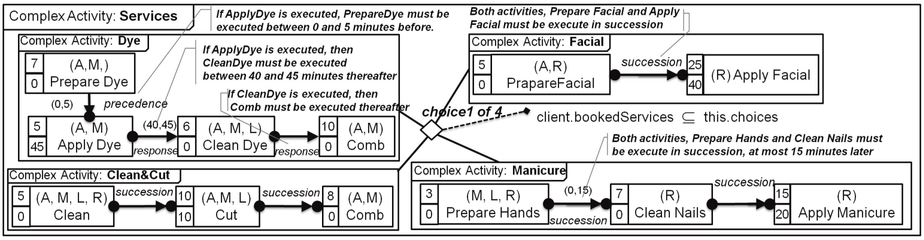

Complex Activity: Services
If ApplyDye is executed, PrepareDye must be executed between 0 and 5 minutes before.
Complex Activity: Dye
7
0
(A,M,)
Prepare Dye
(0,5)
precedence
If ApplyDye is executed, then CleanDye must be executed between 40 and 45 minutes thereafter
If CleanDye is executed, then Comb must be executed thereafter
5
45
(A, M)
Apply Dye
(40,45)
response
6
0
(A, M, L)
Clean Dye
response
10
0
(A,M)
Comb
Complex Activity: Clean&Cut
5
0
(A, M, L, R)
Clean
succession
10
10
(A,M, L)
Cut
succession
8
0
(A,M)
Comb
Both activities, Prepare Facial and Apply Facial must be execute in succession
Complex Activity: Facial
5
0
(A,R)
PrapareFacial
succession
25
40
(R) Apply Facial
choice1 of 4
client.bookedServices ⊆ this.choices
Both activities, Prepare Hands and Clean Nails must be execute in succession, at most 15 minutes later
Complex Activity: Manicure
3
0
(M, L, R)
Prepare Hands
(0,15)
succession
7
0
(R)
Clean Nails
succession
15
20
(R)
Apply Manicure

Fig. 6. ConDec-R Model for the Services Offering

be overcome. Moreover, since multi-objective optimization is considered, several important objectives (i.e., minimizing waiting times for clients and maximizing profit) can be optimized. Furthermore, due to the high expressiveness of ConDec-R, all the constraints which are given in the scenario can be specified

Scenario Details[7]**:** The beauty salon offers various services like dye, clean&cut, manicure and facial services. It requires its clients to make appointment calls to know how many clients are coming as well as the booked services. There are several full-time employees: Amparo (A), Rosa (R), Lisset (L) and Marta (M). Each employee has different skills, and hence some activities can be performed by certain employees only. For all activities which are performed in the salon, the director knows the average estimated duration, the profit which is obtained after its execution, and the employees which can execute that activity. The salon director wants to plan and schedule a working day with several clients taking the following considerations into account:

1. The profit (P) of the resulting working plan has to be maximized (objective function 1).
2. The waiting time (WT) of the clients has to be minimized and distributed uniformly among all the clients (objective function 2):
   $WT = \frac{\sqrt{\sum_{c \in C}((s.endT(c) - s.startT(c)) - (\sum_{b \in s.served(c)} b.estimate))^2}}{C.size}$, where C is the set of clients, s is the considered solution, $s.startT(c)$ and $s.endT(c)$ are the times when the client c starts and finishes in solution s, $s.served(c)$ is the set of services which are applied to c, and $b.estimate$ is the estimated duration for service b.

[7] For the sake of clarity, the considered scenario is a subset of the actual beauty salon, i.e., the salon offers more services and has more employees.

3. The employees can offer some additional services to the client directly in the salon, and the client can accept or refuse. However, these additional services should only be proposed if this leads to optimized plans.

ConDec-R Specification: Typically, as illustrated in Fig. 5, a client visit starts with the reception in the beauty salon. After that, the staff applies some services to the client and, finally, the client is charged. Activity *Services* is composed of other activities[8] (i.e., dye, clean&cut, facial and manicure, cf. Fig. 5), while *Reception* and *Charge* are BP activities (cf. Def. 6). For each BP activity two attributes are considered: (1) estimated activity duration, and (2) profit which is obtained after executing the activity. Moreover, the set of alternative resources which can perform the BP activity is also included. For example, activity *Reception* has an estimated duration of 1 minute and a profit of 0, and can be performed by A, R, M or L. Notice that each instance created from the model of Fig. 5 represents one client visiting the beauty salon. The current problem, however, deals with N clients (represented by the Existence constraint of Fig. 5, stated by the label N) which come to the salon at different times and with different bookings during a working day.

The data perspective also appears in Fig. 5 (cf. Def. 7, Sect. 2). The *ClientData* includes all the information which is related to the client bookings, and consists of: (1) $clientName$, (2) $bookedServices$, which represents the mandatory services that the salon staff has to cover, and (3) $appointmentTime$, which is the time when the client is supposed to arrive at the salon. Through the data perspective, it is possible to model that activity *Reception* cannot start before the client appointment time (cf. Fig. 5). Moreover, a data constraint is used (in conjunction with the choice constraint) to ensure that all services the client has added to her booking are selected, i.e., the generated plans will always include the desired services (cf. Fig. 6).

Generating the Optimized Plans: Given a ConDec-R model $CR = (Acts, Data, C_{BP}, AvRes, OFs)$ for the beauty salon problem, where $Acts$, $Data$, C_{BP} and $AvRes$ are shown in Figs. 5 and 6, and OFs are described in scenario details, the tool generates multi-objective optimized enactment plans which consider both maximizing profit and minimizing waiting time. In this way, the tool suggests: a resource for executing each activity, the start and end time of the activities, and the services which will be offered to each client (i.e., services which were not booked by the client)[9]. The generated optimized plans are then used to support the salon director in managing the working day in an optimized way.

6 Empirical Evaluation

Purpose: The purpose consists of analyzing our proposal in the generation of multi-objective optimized enactment plans from ConDec-R specifications.

[8] Though it is not addressed in the paper, in a similar way to PSL [20], ConDec-R allows hierarchical modelling (i.e., complex activities aggregate activities).

[9] As an example, two Pareto optimal plans (cf. Def. 5) for the beauty salon problem can be found at `http://regula.lsi.us.es/MOPlanner/PlansBeautySalon.pdf`

Objects: The empirical evaluation is based on the beauty salon example described in Sect. 5. This example has been selected for the evaluation, since it is not only a real-world process, but also includes a representative set of ConDec-R templates. The empirical evaluation considers different problems which are obtained by varying the client data of the ConDec-R model of the beauty salon problem. Therefore, each generated problem includes the same activities, relations and resources, but differs in the number of clients (N), their booked services (S), and their appointment times (T). Considering the information which is provided by the salon director, i.e., a client typically books one or two services, we consider values $\{1, 1.5, 2\}$ for the average number of booked services of the clients (i.e., NS). Based on this information, to average the results over a collection of randomly generated ConDec-R models, 30 instances are randomly generated for each pair $< N, NS >$ by varying S and T[10].

Independent Variables: For the empirical evaluation, (1) the number of clients (i.e., N), (2) the average number of booked services for each client (i.e., NS), and (3) the objective function which is selected to be optimized (i.e., OF), are taken as independent variables[11].

Response Variables: The suitability of our approach is tested regarding: (1) the average value of the objective functions which are obtained (i.e, average waiting time (WT) and profit (P)), and (2) the percentage of Pareto optimized solutions within the total number of solutions which are obtained (PS).

Experimental Design: For the model of the beauty salon problem, 270 instances are generated considering different values of N (3 values), NS (3 values), and the automatic generation of T and S (30 problem instances). For each instance[12], 4 searches (i.e., the first 2 searches are executed by optimizing each objective function, and the second 2 searches are executed constraining the values of the other objective function) are executed to compound a diversified and representative Pareto front with at least 4 points. The response variables are then calculated by considering the average values for the 30 problem instances.

Experimental Execution: For the experiments, the constraint-based search algorithm is run until a 5-minute CPU time limit is reached. It is run on a Intel(R) Xeon(R) CPU E5530, 2.40GHz, 8GB memory, running Debian 6.0.3. In order to solve the constraint-based problems, the developed algorithms have been integrated in the system Comet [8].

Experimental Results and Data Analysis: Table 1 shows the values which are obtained for the different response variables. For each problem (specified by N, NS and OF), the average values for the response variables for the 30

[10] The set of problems which are used for the empirical evaluation is available at `http://regula.lsi.us.es/MOPlanner/ObjectsBeautySalon.zip`

[11] Since the ε -constraint method is used, one objective function is optimized while the other one is constrained.

[12] Notice that, for each ConDec-R problem, the corresponding CSP has to deal with more than 50 CSP variables for each client (cf. Sect. 4).

Table 1. Average values related to the experimental execution

Problem			Unconstrained objective			Constrained objective		
N	NS	OF	WT(m)	P(€)	PS(%)	WT(m)	P(€)	PS(%)
10	1	WT	0	361.5	81.4	1.5	410	70
10	1	P	5.9	613.9	65.1	3.2	512.8	59.5
10	1.5	WT	0.6	461.5	80.7	2.3	531.2	70.2
10	1.5	P	8.5	712.8	61.4	5.1	609.5	53.1
10	2	WT	1.6	501.4	76.9	2.8	581.4	57.1
10	2	P	8	787.1	54.7	5.1	688.2	41.3
15	1	WT	1.5	500.5	82.4	4.7	567.1	69.5
15	1	P	11.9	772.6	68.1	7.4	686.7	55.6
15	1.5	WT	1.5	524.8	74	5.8	635.9	51.5
15	1.5	P	11.4	850.2	61.4	7.3	750.8	38.2
15	2	WT	1.9	721.2	67.5	5.7	799.2	40.6
15	2	P	12.1	915	54.2	6.1	856	32.9
20	1	WT	2.2	526.4	71.8	2.7	630.5	54.3
20	1	P	10.2	845.2	59.6	6.5	698.2	42.7
20	1.5	WT	2.3	790.2	68.5	4.9	819.9	50.7
20	1.5	P	10.8	924.4	44.9	5.2	873.7	40.4
20	2	WT	9.1	1045.4	61.2	11.3	1061.1	52.3
20	2	P	15.5	1070.3	44.9	10.8	1060.7	31.4

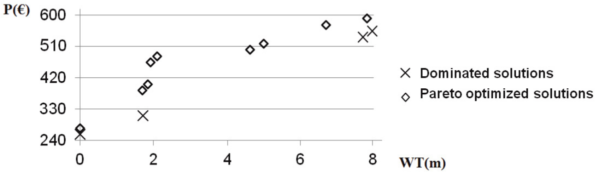


Fig. 7. Solutions which are found for the beauty salon problem for $N = 10$, and $NS = 1$

randomly generated problem instances are shown, i.e., WT, P and PS. For the empirical evaluation, the search algorithm (cf. Sect. 4) is executed in two phases: (1) optimizing one objective function without constraining the other one, i.e., two searches are executed (column Unconstrained Objective), and (2) optimizing one of the objective functions by constraining the other one[13], i.e., two searches are executed (column Constrained Objective). Notice that when OF = WT the constrained objective is P, and vice versa.

As expected, when optimizing one objective function (i.e., WT or P), the confronted objective takes worse values than in the cases in which it is the objective function to be optimized. Moreover, for most problems Pareto optimized

[13] Each objective function is constrained to be better than the average of the two values which are obtained for that function in the phase 1.

solutions could be obtained in average in more than 50% of the time (cf. column PS), which means that a representative Pareto front can be depicted. In general, our results show that constrained problems are harder to solve than unconstrained problems since some CSP variables are more constrained. However, the solutions which are obtained typically present more balanced values for the objective functions.

In Fig. 7 the set of solutions which are found during the search process for a concrete problem of $N = 10$ and $NS = 1$ is depicted. As can be seen, 9 of these solutions are Pareto optimized (cf. Def. 5). The user can choose which of these plans are more valuable according to the relative importance of the two criteria.

7 Discussion and Limitations

The manual specification of BP models, which are traditionally specified through an imperative language, can consume great quantity of resources, cause failures, and lead to non-optimized models, resulting in a very complex problem [10]. We propose specifying processes in a declarative way to facilitate the human work in scenarios with high variability which allow several ways to execute the BP. In addition, the current approach allows modelling the considered problems in an easy way, since the considered declarative specifications are based on high-level constraints. With our extension, an increasing expressiveness is provided (compared to [2,14]), and hence more realistic problems can be managed. Furthermore, the optimized enactment plans are generated by P&S all BP activities considering a set of instances,and hence it allows for a global optimization. Moreover, the automatic generation of optimized plans can deal with complex problems of great size in a simple way, as demonstrated in Sect. 6. Therefore, a wide study of several aspects can be carried out by simulation. As mentioned, the generated plans can be used, among others, for assisting users during flexible process execution to optimize performance through recommendations [3].

However, our approach also presents a few limitations. First, the business analysts must deal with a new language for the constraint-based specification of BPs, thus a period of training is required to let them become familiar with ConDec-R. Secondly, the optimized BP models are generated by considering estimated values for the number of instances, activity attributes and resource availability, and hence our proposal is only appropriate for processes in which these values can be estimated. However, P&S techniques can be applied to replan the activities in runtime by considering the actual values of the parameters [3].

8 Related Work

In the current work, we significantly improve and extend the proposals presented in [2,14] by considering multi-objective optimization, choice [19], temporal [18,29], and data constraints [18], and alternative resources. Hence, more realistic problems and more expressive specifications can be managed. We are not aware of any other approaches for generating enactment plans from declarative specifications, however, there exist some further proposals which could be extended in such direction [19,18,15,17]. Specifically, [19] proposes the generation of a non-deterministic finite state automaton from constraint-based specifications which represents exactly all traces that satisfy the constraints. However, the big disadvantage following such an approach would be that the process of generating the automaton from the declarative specifications is NP complete, and, unlike the proposed approach, no heuristic is used. Additionally, CLIMB [18] could be used to generate quality traces from declarative specifications, and calculate its values for different objective functions. Then, the best traces could be selected. Unlike the proposed approach, [18] does neither consider optimality nor resource availabilities. Therefore, these would only cover the planning part of the current proposal, but not the scheduling aspects. The work [15] plans and schedules tasks considering resources and the optimization of one objective function through an integer constraint-based specification. Although [15] presents a similar constraint-based approach, it misses dealing with multi-objective optimization, and does not support high level constraints. In [17], a constraint formalization is proposed to generate variations of an ad-hoc BPMN sub-processes. Unlike our approach, [17] does not consider objective optimization.

Several filtering rules for specialized scheduling constraints have been developed. Specifically, [4,16] model scheduling problems which include alternative and optional tasks respectively, together with their filtering rules. The proposed model and propagation for the optional activities in the current work are very similar to the proposal presented in [16]. However, unlike [4,16], to efficiently manage ConDec-R constraints we developed complex and innovative filtering rules related to the alternating executions of repeated activities together with the variable number of times which these activities are executed.

9 Conclusions and Future Work

In this work, generating optimized BP enactment plans from declarative specifications is proposed to optimize the performance of a process. The generated plans can be used for different purposes, e.g., generating recommendations. We improve and extend a previous work motivated by the requirements described in literature as well as the necessities of the case studies we have conducted. Moreover, the proposed approach is validated through an empirical evaluation. As for future work, we will explore various constraint-based solving techniques and analyze their suitability for the generation of multi-objective optimized plans. Additionally, we intend to consider further resource patterns. Though the developed experiments cover only technical details of the approach (cf. Sect. 6), the results obtained seem promising when being analyzed with the manager. As for future work an empirical study is proposed to be designed to measure the improvements achieved in the business.

Acknowledgements. This work has been partially funded by the Spanish Ministerio de Ciencia e Innovación (TIN2009-13714) and the European Regional Development Fund (ERDF/FEDER).

References

1. Athan, T.W., Papalambros, P.Y.: A note on weighted criteria methods for compromise solutions in multi-objective optimization. Eng. Optim. 27(2), 155–176 (1996)
2. Barba, I., Del Valle, C.: A Constraint-based Approach for Planning and Scheduling Repeated Activities. In: Proc. COPLAS, pp. 55–62 (2011)
3. Barba, I., Weber, B., Del Valle, C.: Supporting the Optimized Execution of Business Processes through Recommendations. In: Proc. BPI (2011) (in press)
4. Bartak, R., Cepek, O.: Incremental propagation rules for a precedence graph with optional activities and time windows. Trans. Inst. Meas. Control 32(1), 73–96 (2010)
5. Brucker, P., Knust, S.: Complex Scheduling (GOR-Publications). Springer-Verlag New York, Inc, Secaucus (2006)
6. Cheng, D., Li, F.Y.: Multiobjective optimization design with pareto genetic algorithm. J. Guid. Control Dyn. 19, 392–397 (1997)
7. Declare. Declare: Declarative Approach to Workflow Management Systems (2011), `http://www.win.tue.nl/declare/` (accessed May 1, 2012)
8. Dynadec. Comet Downloads (2010), `http://dynadec.com/support/downloads/` (accessed May 1, 2012)
9. Ehrgott, M., Gandibleux, X.: Multiobjective combinatorial optimization - theory, methodology, and applications. In: Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys. Int. Series in Op. Res. & Man. Sci, vol. 52, pp. 369–444 (2003)
10. Ferreira, H.M., Ferreira, D.R.: An integrated life cycle for workflow management based on learning and planning. Int. J. Cooper Inform. Syst. 15(4), 485–505 (2006)
11. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)
12. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory and Practice. Morgan Kaufmann, Amsterdam (2004)
13. Haimes, Y.Y., Lasdon, L.S., Wismer, D.A.: On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Trans. on Syst. Man. Cybern. 1, 296–297 (1971)
14. Jiménez-Ramírez, A., Barba, I., Del Valle, C., Weber, B.: OptBPPlanner: Automatic Generation of Optimized Business Process Enactment Plans. In: Proc. ISD. Springer (2012) (in press)
15. Krogt, R., Geraghty, J., Salman, M.R., Little, J.: On supporting lean methodologies using constraint-based scheduling. J. of Scheduling 13, 301–314 (2010)
16. Laborie, P., Rogerie, J., Shaw, P., Vilim, P.: Reasoning with conditional time-intervals. part ii: An algebraical model for resources (2009)
17. Lu, R., Sadiq, S., Governatori, G., Yang, X.: Defining adaptation constraints for business process variants. In: Abramowicz, W. (ed.) BIS 2009. LNBIP, vol. 21, pp. 145–156. Springer, Heidelberg (2009)
18. Montali, M.: Specification and Verification of Declarative Open Interaction Models: a Logic-Based Approach. PhD thesis, Department of Electronics, Computer Science and Telecommunications Engineering, University of Bologna (2009)
19. Pesic, M.: Constraint-Based Workflow Management Systems: Shifting Control to Users. PhD thesis, Eindhoven University of Technology, Eindhoven (2008)
20. Process Specification Language project (1977), `http://www.nist.gov/psl/` (accessed May 1, 2012)

21. R-Moreno, M.D., Borrajo, D., Cesta, A., Oddi, A.: Integrating planning and scheduling in workflow domains. Expert Syst. Appl. 33(2), 389–406 (2007)
22. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems. Springer (2012)
23. Rossi, F., van Beek, P., Walsh, T. (eds.): Handbook of Constraint Programming. Elsevier (2006)
24. Rozinat, A., Wynn, M.T., van der Aalst, W.M.P., ter Hofstede, A.H.M., Fidge, C.J.: Workflow simulation for operational decision support. Data Knowl. Eng. 68(9), 834–850 (2009)
25. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Edmond, D.: Workflow resource patterns: Identification, representation and tool support. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 216–232. Springer, Heidelberg (2005)
26. Salido, M.A.: Introduction to planning, scheduling and constraint satisfaction. J. Intell. Manuf. 21(1), 1–4 (2010)
27. van der Aalst, W.M.P., Schonenberg, M.H., Song, M.: Time prediction based on process mining. Inform. Syst. 36(2), 450–475 (2011)
28. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2007)
29. Westergaard, M., Maggi, F.M.: Looking into the future: Using timed automata to provide a priori advice about timed declarative process models. In: Meersman, R., Panetto, H., Dillon, T., Rinderle-Ma, S., Dadam, P., Zhou, X., Pearson, S., Ferscha, A., Bergamaschi, S., Cruz, I.F. (eds.) OTM 2012, Part I. LNCS, vol. 7565, pp. 250–267. Springer, Heidelberg (2012)