

# Extending Evolutionary Multi-Objective Optimization of Business Process Designs

M.Sc. Thesis of Konstantinos Georgoulakos

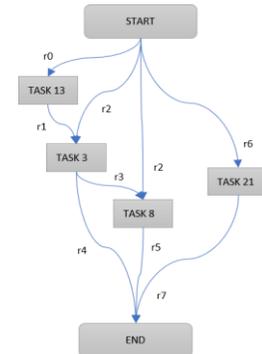
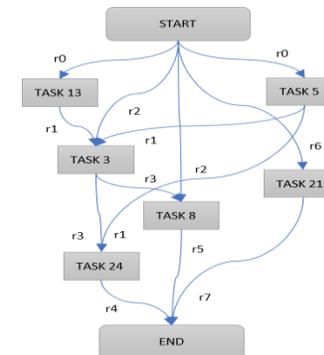
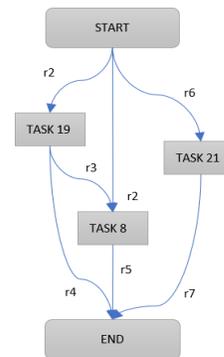


# Business Process Optimization (BPO)

Business process optimization is considered as the problem of constructing feasible business process designs with optimum attribute values such as duration and cost.

No.	Task Name	Input(s)	Output(s)	SDP	SFT
0	Achworks Soap (T\$\$ - Rico Pamplona)	1, 2	3	208	113
1	BAX Global Tracking Service	2, 3	5	219	109
2	CDYNE Death Index	1, 3	4	229	115
3	Credit Card Processor	1, 2	3,4	202	109
4	D&B Business Credit Quick Check	1, 3	4	203	108
5	Drupal authentication	0	1	200	103
6	ecommStats Web Analytics	6	7	218	112
7	Entrust login	0	1	206	103
8	FedEx Tracker	2, 3	5	211	109
9	FedEx / UPS Package Tracking	2, 3	5	224	103
10	FraudLabs Credit Card Fraud Detection	1, 3	4	220	113
11	Google Analytics	6	7	218	107
12	Google Checkout	1, 2	3	206	105
13	GUID Generator	0	1	203	110
14	Internet Payment Systems	1, 2	3	226	105
15	LID login	0	1	222	114
16	OpenID login	0	1	228	100
17	Paypal online payment	1, 2	3	215	102
18	Real Time Check Verification (T\$\$ - Rico Pamplona)	3	4	229	108
19	Rich Payments NET	2	3, 4	208	105
20	SAINTlogin users validation	0	1	219	105
21	Servicetrack	6	7	212	113
22	SmartPayments Payment	2	3	214	105
23	Smartpayments CardValidator	3	4	206	107
24	Strikelron Global Address Verification	1, 3	4	201	105
25	SXIP login	0	1	203	105
26	Typekey authentication service	0	1	224	114
27	UPS Tracking	2, 3	5	225	109
28	VeriSign Payment	1, 2	3	230	103

No.	Resource name	Process I/O
0	Customer account credentials	I
1	Customer account details	
2	Order details	I
3	Payment details	
4	Payment confirmation	O
5	Order tracking status	O
6	Website tracking request	I
7	Website statistics	O



# Why Evolutionary Multi-Objective Optimization?

Evolutionary algorithms (EAs) mimic nature's evolutionary principles to guide the optimization process towards discovering optimal solutions by iteratively growing or developing a population of solutions.

## Characteristics of EAs:

- A population of solution is found instead of a single solution. This enables them to identify a number of optimal solutions in the final population
- The lack of preference towards a specific optimization objective gives them the capability of providing a wide range of optimal solutions that each of them reflects a different trade-off among the optimization objectives

## Benefits for BPO:

- Business process designs that would otherwise be overlooked by a human designer can be discovered
- Evolving a solution over the generations can transform an infeasible process design to a feasible one
- Based on specific objectives, the fittest process design can be determined by evaluating a significant number of alternative designs based on the same process

# Previous approaches

## *Algorithmical approaches to business process design, Hofacker I. and Vetschera R. (2001)*

- The first who applied EAs to BPO problems
- The most comprehensive assumptions for the feasibility of a business process design
- Very weak performance because the feasibility couldn't be maintained in a tightly constrained problem

## *Business Process Optimisation using an Evolutionary Multi-Objective Framework, Vergidis K. (2008)*

- The first who introduced an optimization framework for BPO problems,  $BPO_F$
- Employed the most state-of-the-art Evolutionary Multi-objective Optimization Algorithms (EMOAs)
- Introduced PCA, the process composition algorithm for maintaining and preserving the feasibility during optimization
- Introduced the Degree of Infeasibility (DoI) for measuring the extent to which a process design is infeasible
- Satisfactory results for problems coming from the Service Industry

# Aim & Objectives

## Aim

The aim of this thesis is the improvement and the extension of previous approaches for Evolutionary Multi-objective Business Process Optimization (EMBPO).

## Objectives

- Studying and understanding the previous approaches for BPO
- Reviewing the results of BPO<sub>F</sub> reported by Vergidis
- Improving, developing and extending BPO<sub>F</sub> on the same principles using Python
- Introducing a pre-processing stage for BPO problems
- Proposing a novel design composition algorithm for business processes, PCA-II
- Validation and testing of the proposed framework

# Deliverables

The outcome of this research is a revised and improved version of  $BPO_F$ , the extended business process optimization framework ( $eBPO_F$ ).

## Features of $eBPO_F$

- A pre-processing technique for enhancing the efficiency of the employed EMOAs
- PCA-II, a new process composition algorithm which is suitable for real-world complex problems
- Ease of use
- More efficient I/O
- Better interactivity
- Easy maintenance

# Pre-processing technique

This technique aims for increasing the ability of eBPO<sub>F</sub> to find the optimized design alternatives by removing the tasks that would never be part of such a design.

## Necessity

- The efficiency of EMOAs and the quality of the produced results highly depend on the size of the examined problem
- Business processes can have too many available alternatives for the participant tasks and these in turn can involve many different resources as requirements or as products of their utilization
- The wrong configuration should be easily distinguished from the complex or infeasible problem, especially in cases where eBPO<sub>F</sub> is used as a business process mining tool

## Scope

- Clearing the problem dataset out of tasks that will never be part of the best solution or tasks whose features don't comply with the problem constraints
- Checking if there are tasks in the library that can meet the process requirements e.g. process inputs and outputs

# Pre-processing sub-processes

## Check Global Inputs & Outputs

- Tasks that their set of input resources is a superset of the set of the process output resources must be removed.
- Tasks that their set of output resources is a subset of the set of the process input resources must be removed

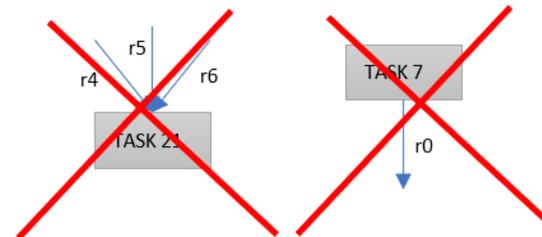
### Check Global Inputs & Outputs

- Iterate on the Valid Tasks
  - If the output resources of the valid task iterated is a subset of the set of the process input resources
    - Set the valid task iterated as invalid from now on
  - If the valid task iterated is still valid
    - If the set of the process output resources is a subset of the input resources of the valid task iterated
      - Set the valid task iterated as invalid from now on

### Inputs & Outputs

- Problem Specs

Process Inputs: [0, 2]  
Process Outputs: [4,5]



# Pre-processing sub-processes

## Check Task Inputs

- Tasks that require an input resource which is not produced by any other task or the set of process input resources must be removed

### Check Task Inputs

- Number of Initial Valid Tasks = 0
- Number of Updated Valid Tasks = number of Valid Tasks
- While number of Initial Valid Tasks  $\neq$  number of Updated Valid Tasks
  - Number of Initial Valid Tasks = number of Updated Valid Tasks
  - Iterate on the Valid Tasks
    - Iterate on the input resources of the valid task iterated
      - If the input resource iterated is a global one
        - ❖ Continue with the next input resource of the valid task iterated
      - If the input resource iterated is not an output resource of another valid task
        - ❖ Set the valid task iterated as invalid from now on
        - ❖ Exit the input resources loop for the previously valid task iterated
  - Number of Updated Valid Tasks = number of Valid Tasks

### Inputs & Outputs

- Problem Specs

The sub-process is repeated until no more valid tasks become invalid.

If a valid task becomes invalid, there may be some other formerly valid tasks, whose some of their input resources could only be obtained by the output resources of that currently invalid task, thus these must become invalid too.

# Pre-processing sub-processes

## Check Task Outputs

- Tasks that none of their output resources is required as input by another other task or the set of the process output resources must be removed

### Check Task Outputs

- Number of Initial Valid Tasks = 0
- Number of Updated Valid Tasks = number of Valid Tasks
- While number of Initial Valid Tasks != number of Updated Valid Tasks
  - Number of Initial Valid Tasks = number of Updated Valid Tasks
  - Iterate on the Valid Tasks
    - Iterate on the output resources of the valid task iterated
      - If the output resource iterated is a global one
        - ❖ Exit the output resources loop of the valid task iterated
      - If the output resource iterated is an input resource for another valid task
        - ❖ Exit the output resources loop of the valid task iterated
      - Else If the output resource iterated is the last one
        - ❖ Set the valid task iterated as invalid from now on
  - Number of Updated Valid Tasks = number of Valid Tasks

### Inputs & Outputs

- Problem Specs

The sub-process is repeated until no more valid tasks become invalid.

If a valid task becomes invalid, there may be some other formerly valid tasks, whose all their output resources could only be given to satisfy the input resources of that currently invalid task, thus these must become invalid too.

# Pre-processing sub-processes

## Check Categories

- Tasks that have the same input and output resources so that they can be classified, and are dominated by other tasks of the same classification according to their attribute values and the kind of the examined problem must be removed

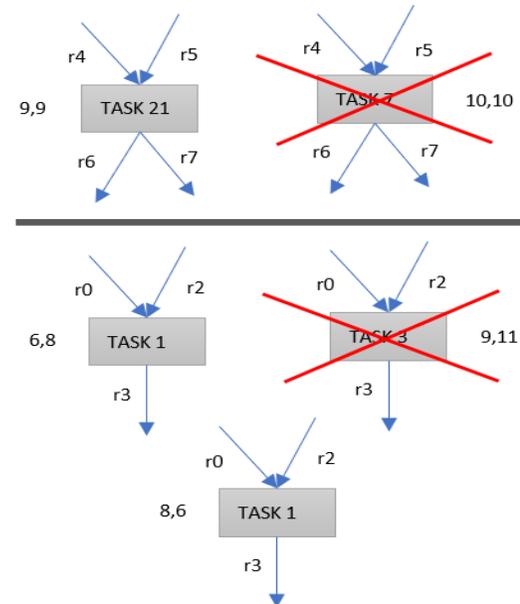
### Check Categories

- Initialize Categories as an empty dictionary
- Mark all Valid Tasks as not belonging to a specific category yet
- Iterate on Valid Tasks
  - If the valid task iterated does not belong to a specific category
    - Mark the valid task iterated as belonging to a category
    - Add a new category to the Categories dictionary
    - Add the valid task iterated to the newly created category
    - Iterate on the rest Valid Tasks
      - If the inner valid task iterated does not belong to a specific category
        - ❖ If the outer valid task iterated has exactly the same input and output resources with the inner valid task iterated
          - Mark the inner valid task iterated as belonging to a category
          - Add the inner valid task iterated to the newly created category
- Iterate on Categories
  - If the category iterated has more than one Valid Tasks
    - Get the Problem Type e.g. ("MIN-MIN", "MIN-MAX", "MAX-MIN", "MAX-MAX")
    - Iterate on the tasks of the category iterated
      - If the category task iterated is still valid
        - ❖ Iterate on the rest tasks of the category iterated
          - If the inner category task iterated is still valid
            - ✓ If the inner category task iterated dominates the outer category task iterated according to the Problem Type
              - @ Set the outer category task iterated as invalid from now on
              - @ Exit the loop on the rest tasks of the category iterated
            - ✓ Else If the outer category task iterated dominates the inner category task iterated according to the Problem Type
              - @ Set the inner category task iterated as invalid from now on

#### Inputs & Outputs

- Problem Specs

### Problem Type: MIN-MIN



# Pre-processing sub-processes

## Check Global Availability

- Examine whether the problem specification is valid for the given dataset or not

### Check Global Availability

- Scenario Validity = False
- Iterate on the Process Input Resources
  - If the process input resource can be utilized by a valid task
    - Scenario Validity = True
    - Exit the Process Input Resources loop
- If the scenario is valid
  - Iterate on the Process Output Resources
    - If the process output resource cannot be produced by any valid task
      - Scenario Validity = False
      - Exit Process Output Resources loop
- Return Scenario Validity

#### Inputs

- Problem Specs

#### Outputs

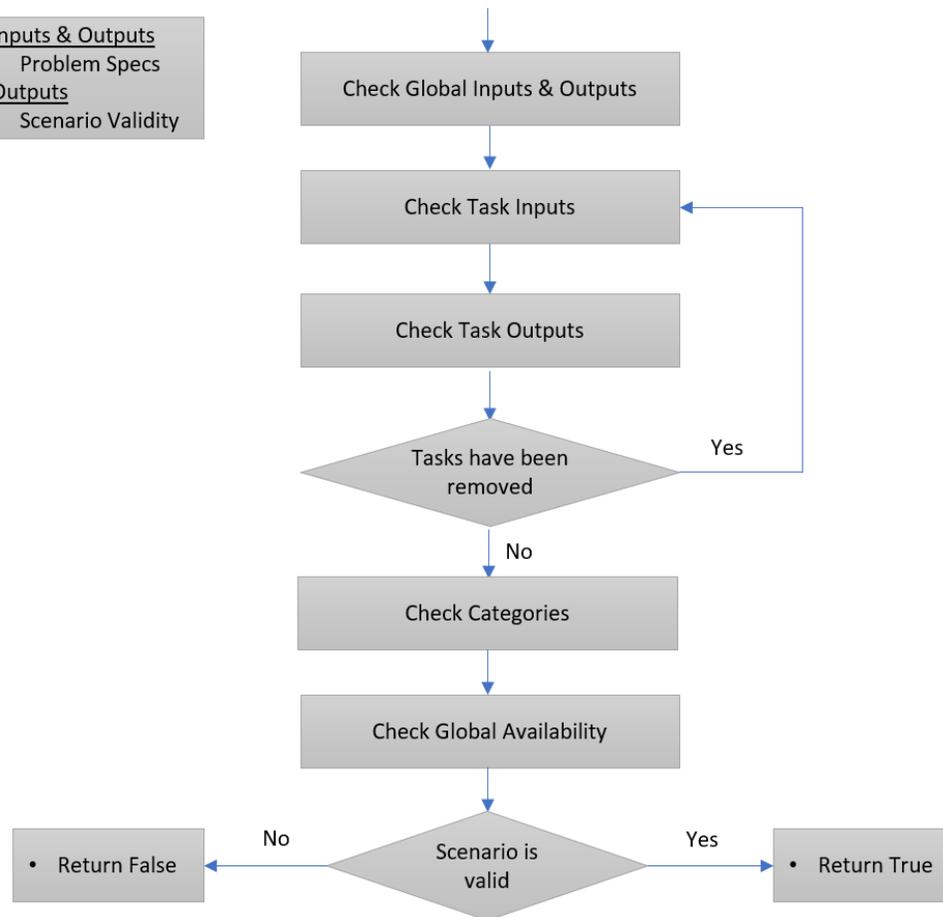
- Scenario Validity

1. At least one of the process input resources must be utilized during the process by the participating tasks
2. All process output resources must be produced during the process by the participating tasks

# Pre-processing algorithm

## Inputs & Outputs

- Problem Specs
- Outputs
- Scenario Validity



- At the beginning, all the tasks in library are considered as valid
- Each sub-process operation is performed only on the remaining valid tasks of the library of tasks
- At the end, a Boolean variable is returned to signify whether the main optimization operation should be performed or not

The sub-processes “Check Task Inputs” and/or “Check Task Outputs” are repeated until no more valid tasks become invalid.

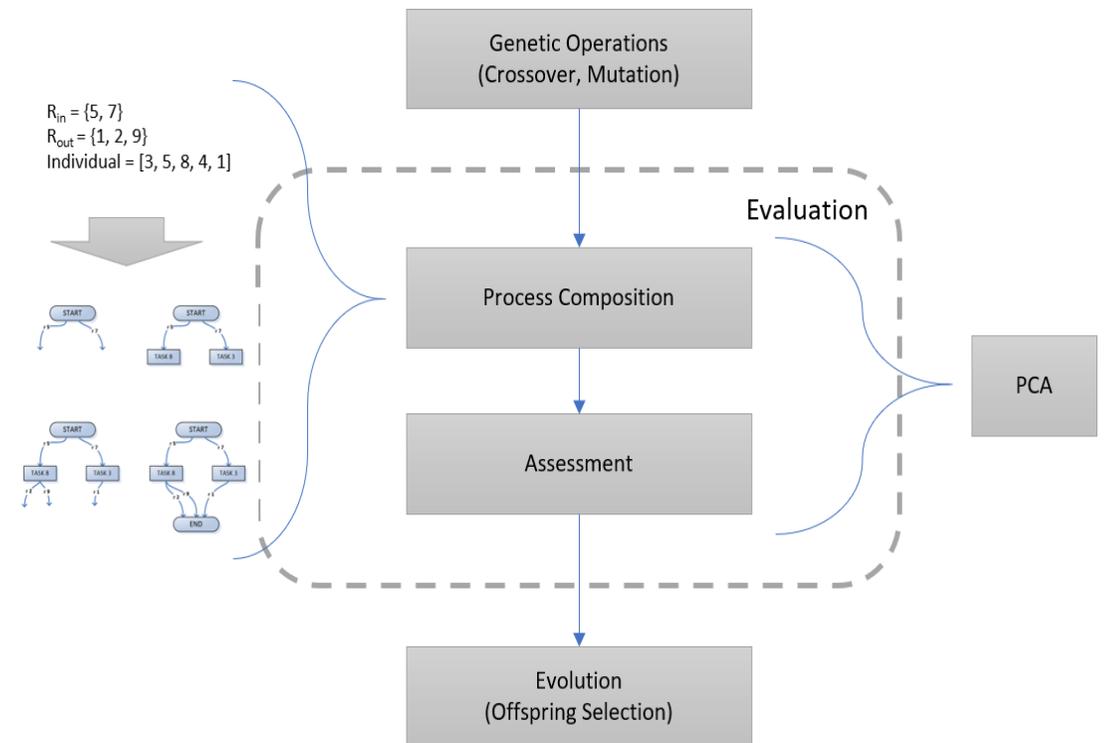
If valid tasks become invalid by “Check Task Inputs” and/or “Check Task Outputs”, there may be some other formerly valid tasks, whose all their output resources could only be given to satisfy the input resources of currently invalid tasks and/or some of their input resources could only be obtained by the output resources of currently invalid tasks and thus, these must become invalid too.

# Process Composition

The process composition is the step where the quantitative representation of a business process is transformed to the visual one.

PCA attempts to compose a feasible process diagram for every solution in the population using its quantitative representation and considering the process requirements.

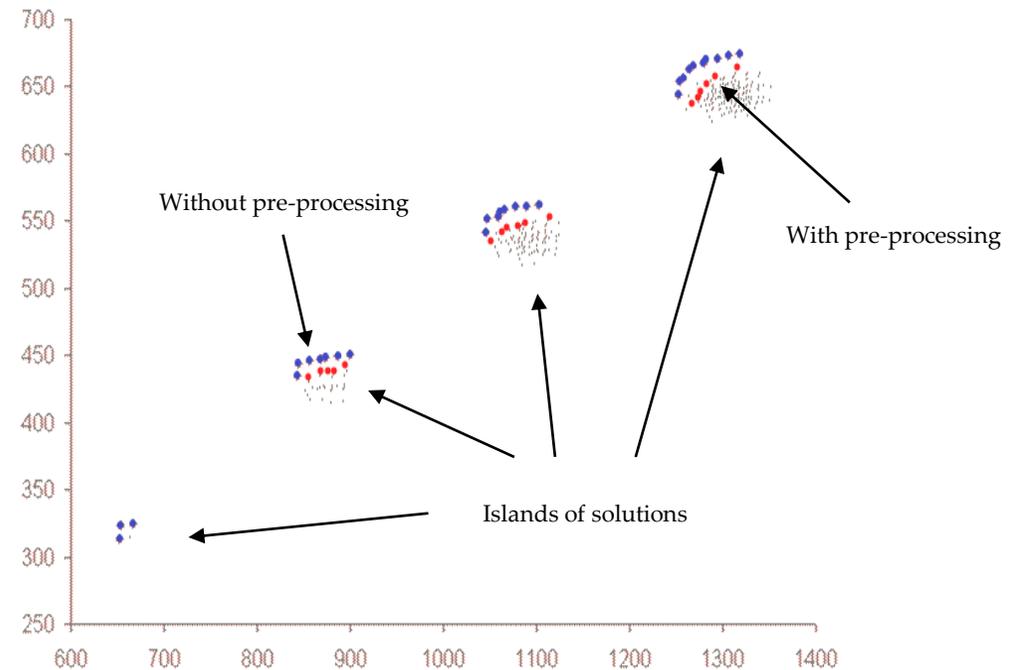
PCA is also responsible for the assessment of every solution in the population in terms of examining whether the produced process design for that solution is feasible according to the problem specification or not, and properly updating its attributes values that will be used in the evolution of the population afterwards.



# Need for PCA-II

## Low quality results of $BPO_F$ with pre-processing

- The pareto front of the solutions wasn't even the same with that without pre-processing, but it was always worse in all runs conducted
- For the problem "Scenario B: Sales forecasting",  $BPO_F$  was unable to find a whole island of solutions



# Need for PCA-II

## Weakness of PCA

PCA cannot meet the requirements of a generic real life business process design regarding feasibility.

## Assumptions of PCA regarding feasibility

1. All process input resources are utilized by one or more tasks participating in the process design
2. All process output resources are produced by one or more tasks participating in the process design
3. Each task in the design is connected either with the process inputs, the process outputs or another task in the design

The third constraint is too loose and can ensure the feasibility of processes whose tasks require only one input resource, but generally the participating tasks in a process design may need more than one input resources to be executed.

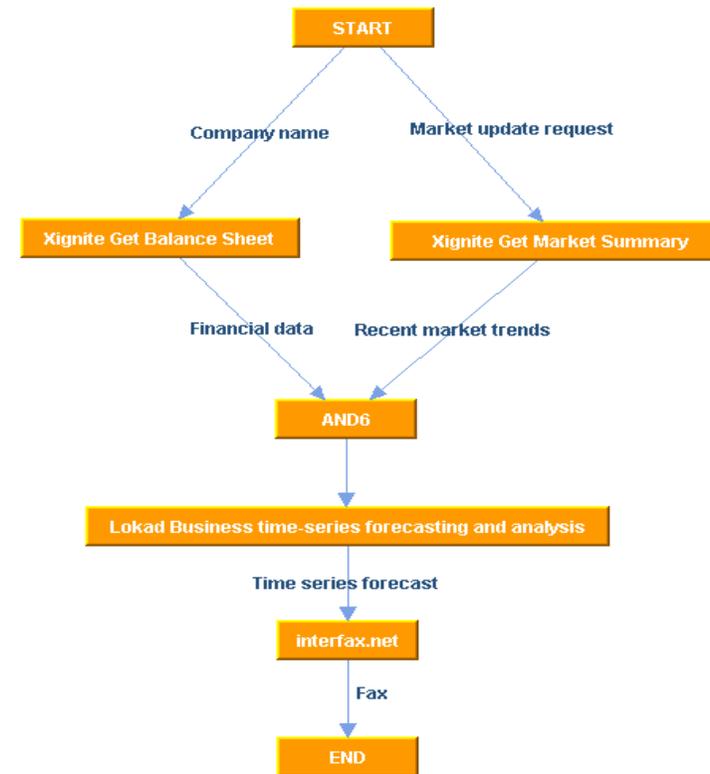
# Need for PCA-II

No.	Task Name	Input(s)	Output(s)	SDP	SFT
0	D&B Business Verification	3,1	0,5	206	103
1	Fax.com	8,2	4	220	103
2	Gale Group Business Information	3,0	5	223	106
3	Gale Group Business Intelligence	3,1	0,5	229	113
4	GraphMagic's Graph & Chart Web Service API	5,8	2	203	107
5	interfax.net	8,2	4	222	113
6	Lokad Business time-series forecasting and analysis	5,7	8	228	110
7	Midnight Trader Financial News	6,3	7	217	101
8	Strikelron Company Search	3	3,0	230	114
9	Strikelron Get Business Prospect	3,1	0,5	205	110
10	Strikelron Lookup Business	3	3,0	201	110
11	Wall Street Horizon Real-Time Company Earnings	3,1	5	210	105
12	Xignite Get Balance Sheet	3,1	5	216	112
13	Xignite Get Chart Url	8	2	228	110
14	Xignite Get Chart Url Preset	8	2	228	101
15	Xignite Get Growth Probability	5,7	8	215	109
16	Xignite Get Market News Headlines	6	7	221	114
17	Xignite Get Market Summary	6	7	203	112
18	Xignite Get Topic Chart	3,5,7	8	218	112
19	Xignite Get Topic Data	3,5,7	8,2	222	109

Task library of scenario B

No.	Resource name	Process I/O
0	Business details	
1	Business query	
2	Chart / graph	
3	Company name	I
4	Fax (on-line)	O
5	Financial data	
6	Market update request	I
7	Recent market trends	
8	Time-series forecast	

Available Resources for scenario B

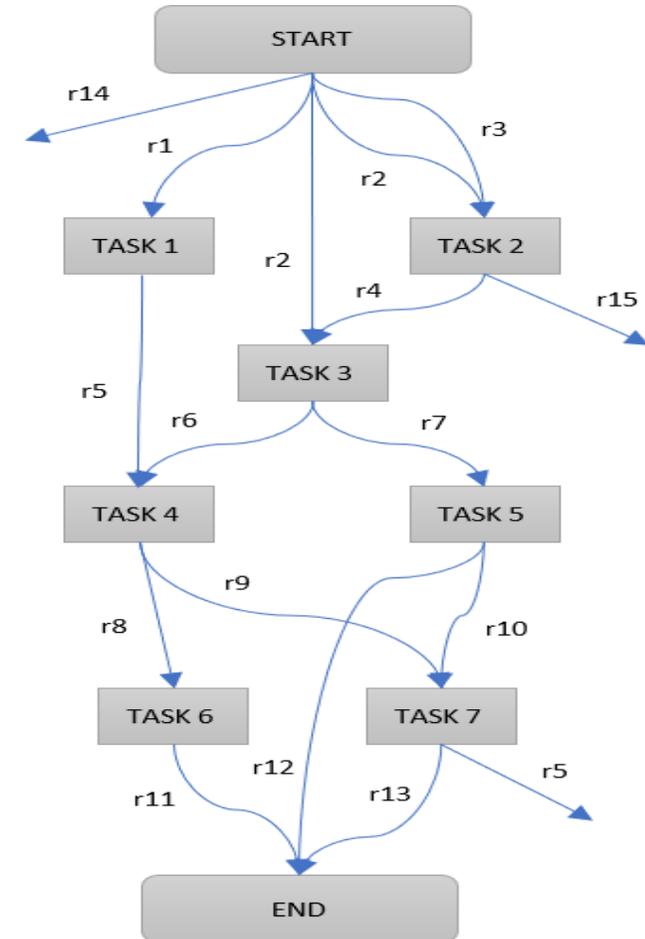


# PCA-II

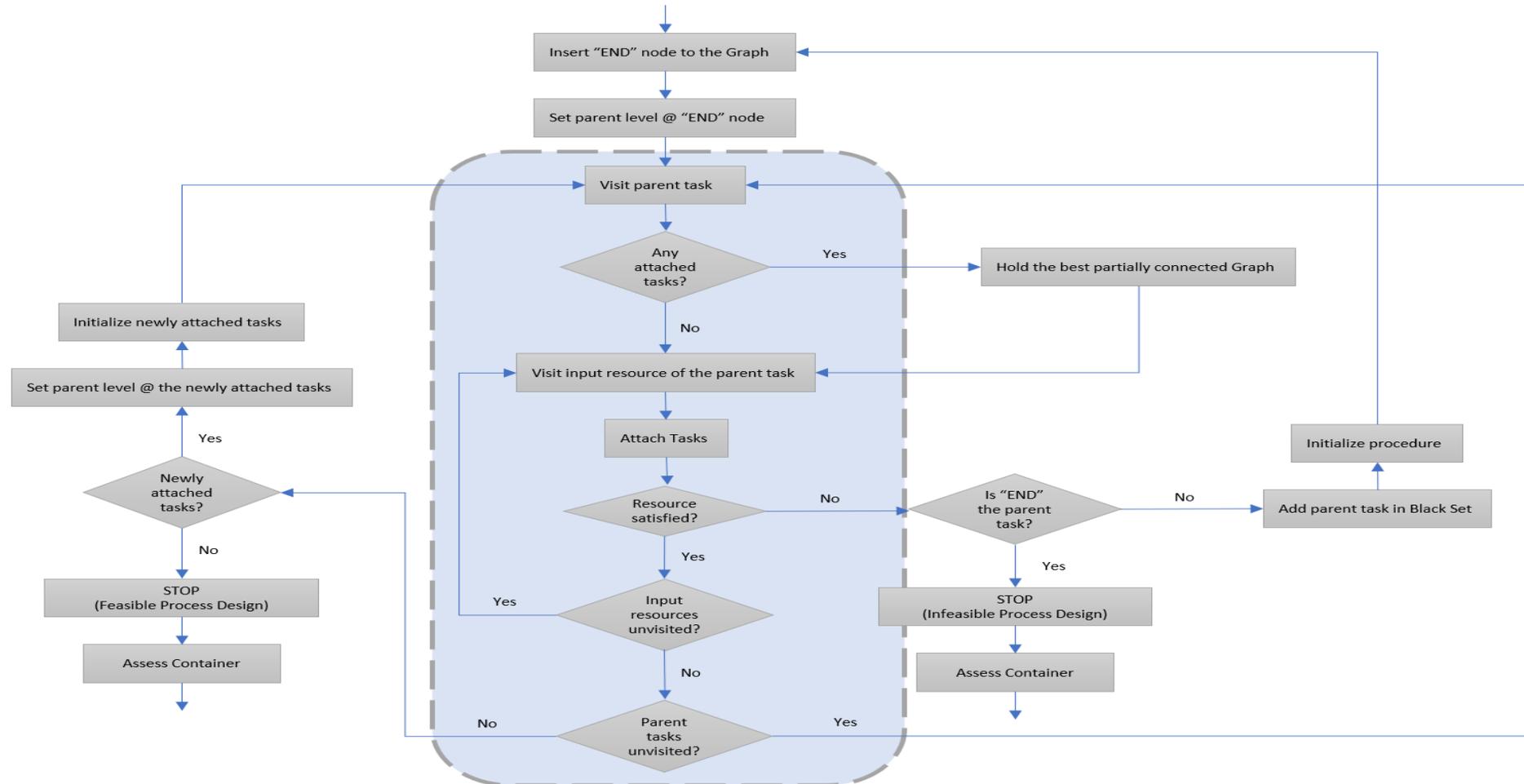
## Assumptions of PCA-II regarding feasibility

1. All process output resources are produced by one or more participating tasks in the design
2. All input resources of the participating tasks in the process design are satisfied either by the output resources of their preceding tasks in the design or by the process input resources
3. All participating tasks in the process design provide at least one of their output resources for satisfying a process output resource or an input resource of a succeeding task in the design

These rules are rather well-rounded to capture the needs of a wide range of business process domains



# Composition approach of PCA-II



# Business process representation of PCA

- PCA assesses the feasibility of the solutions in the population in a way that at the end, the population will contain the alternative most optimized process designs for the process requirements.
- Every solution in the population starts with the maximum numbers of tasks that a solution can have
- Afterwards, through generations, PCA modifies the solutions by adding or removing tasks
- Finally, every feasible solution contains only the participating tasks

# Business process representation of PCA-II

- Every solution in the population is considered as a container of tasks and PCA-II makes attempts to compose a feasible process design using some of those tasks
- The assessment of the containers involves the ability of PCA-II to compose a feasible process design with their tasks and the attribute values of those tasks
- At the end, the population must contain the most promising containers for composing a feasible optimized process design and this is the aim of the evaluation performed in the new algorithm
- At the whole optimization phase, all containers in the population, have the same predefined number of tasks and no change is performed to the tasks of a container except for the genetic operators, crossover and mutation

# Business process representation of PCA-II

- PCA-II examines every container in the population whether it contains a feasible process design or not
- In both cases, after the composition attempts, a list with the best partially connected tasks in the design and the graph of this design will be used for the assessment of the examined container
- In case of process feasibility, the graph corresponds to the first feasible process design composed by the algorithm and the list contains all participating tasks
- Otherwise, the graph corresponds to the best possible elaboration of the design with the tasks in the container before one of those tasks cannot be satisfied for one of its input resources and the list contains the participating tasks in that design

# Business process representation of PCA-II

- The tasks in the list are added in the same order which are attached to the graph during the composition attempt
- This list and the graph for a feasible design are added in "Hall of Fame" where all feasible designs found in the execution of  $eBPO_F$ , are stored
- The order which the tasks are presented in the container, matters; the searching for a task in the container that satisfies the examined input resource, is always performed from left to right

PCA-II ensures that:

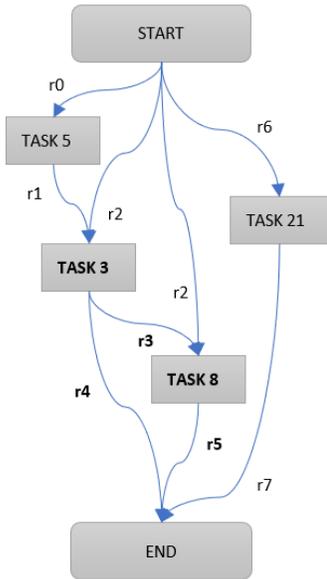
1. Given a container of tasks, it will always construct the same "most feasible" design during its composition attempts
2. The results of  $eBPO_F$  execution can be reviewed. The process composition of the tasks of the list-container accompanying a feasible solution leads to the same process design produced by  $eBPO_F$

# Business process representation of PCA-II

2	8	19	3	5	21	4	22	0	10
---	---	----	---	---	----	---	----	---	----

2	8	19	3	5	21	7	22	0	10
---	---	----	---	---	----	---	----	---	----

3	8	21	5
---	---	----	---

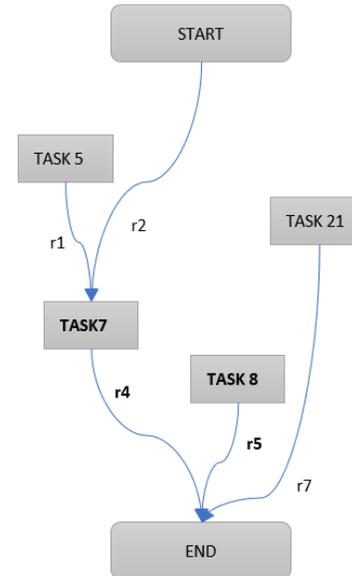


**FEASIBLE**



2	8	19	7	5	21	4	22	0	10
---	---	----	---	---	----	---	----	---	----

7	8	21	5
---	---	----	---

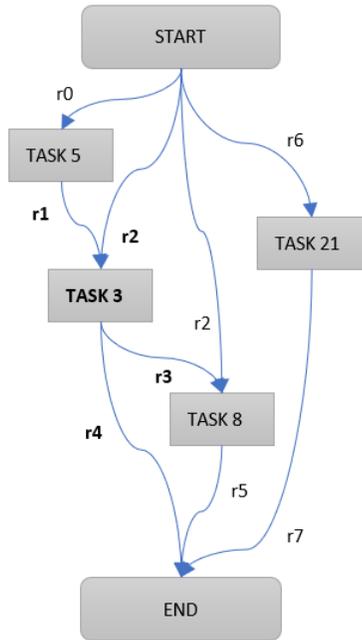


**INFEASIBLE**

# Business process representation of PCA-II

2	8	19	3	5	21	6	22	0	10
---	---	----	---	---	----	---	----	---	----

3	8	21	5
---	---	----	---

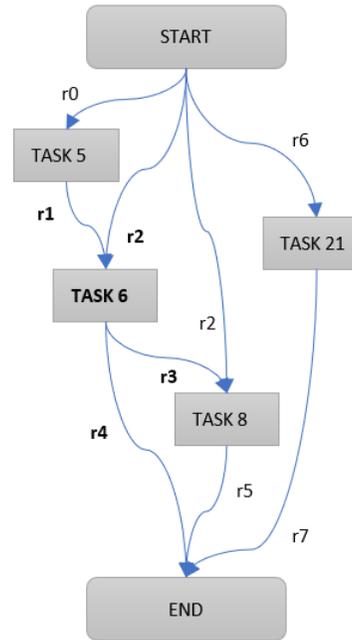


**FEASIBLE**

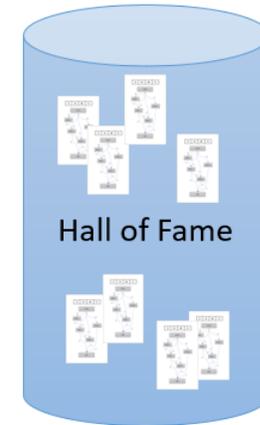


2	8	19	6	5	21	3	22	0	10
---	---	----	---	---	----	---	----	---	----

6	8	21	5
---	---	----	---



**FEASIBLE**



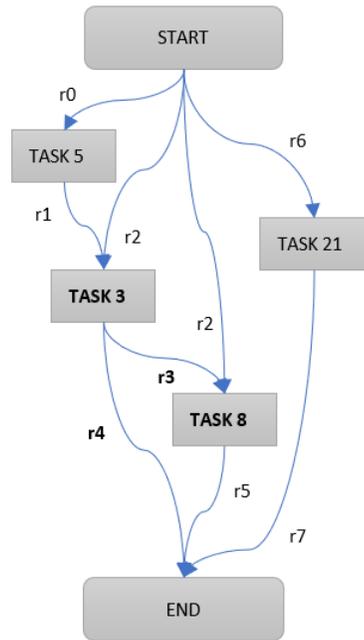
# Business process representation of PCA-II

2	8	19	3	5	21	7	22	0	10
---	---	----	---	---	----	---	----	---	----

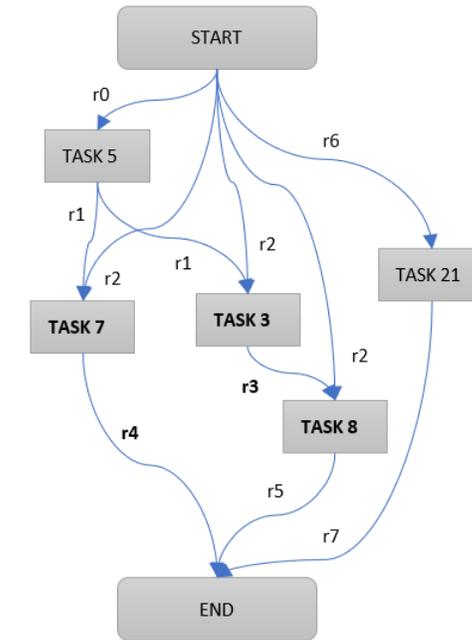
2	8	19	7	5	21	3	22	0	10
---	---	----	---	---	----	---	----	---	----

3	8	21	5
---	---	----	---

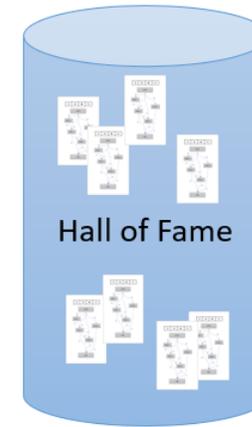
7	8	21	5	3
---	---	----	---	---



**FEASIBLE**



**FEASIBLE**



# Problem Formulation in eBPO<sub>F</sub>

Parameter	Description	Parameter	Description
n	Number of tasks in the library	N	Set of the n tasks
n <sub>d</sub>	Number of tasks in the design	N <sub>d</sub>	Set of the n <sub>d</sub> tasks
n <sub>c</sub>	Number of tasks in the container	N <sub>c</sub>	Set of the n <sub>c</sub> tasks
n <sub>min</sub>	Minimum number of tasks in the design	n <sub>max</sub>	Maximum number of tasks in the design
r	Number of available resources	Dol	Degree of Infeasibility
r <sub>in</sub>	Number of process input resources	R <sub>in</sub>	The set of process input resources
r <sub>out</sub>	Number of process output resources	R <sub>out</sub>	The set of process output resources
t <sub>in</sub> <sup>i</sup>	Number of input resources of task i	T <sub>in</sub> <sup>i</sup>	Set of input resources of task i
t <sub>out</sub> <sup>i</sup>	Number of output resources of task i	T <sub>out</sub> <sup>i</sup>	Set of output resources of task i
p	Number of task/process attributes	TA <sub>i</sub>	Attribute values of task i
		PA	Process attribute values
c	Number of container attribute values	CA	Container attribute values

For a business process design with a set of n<sub>d</sub> tasks and p process attributes:

Minimize/maximize PA<sub>i</sub>, i ∈ {1, 2, ..., p}

Subject to:

1. Dol = 0
2. n ≥ n<sub>max</sub> ≥ n<sub>d</sub> ≥ n<sub>min</sub> ≥ 0
3. n ≥ n<sub>c</sub> ≥ 0
4. r ≥ r<sub>in</sub>, r<sub>out</sub>, t<sub>in</sub><sup>i</sup>, t<sub>out</sub><sup>i</sup> > 0, i ∈ {1, 2, ..., n<sub>d</sub>}
5. p ≥ 2
6. c = p + 1

$$PA_j = \sum_{i=1}^{n_d} TA_{ij}$$

$$CA_j = \sum_{i=1}^{n_c} TA_{ij}$$

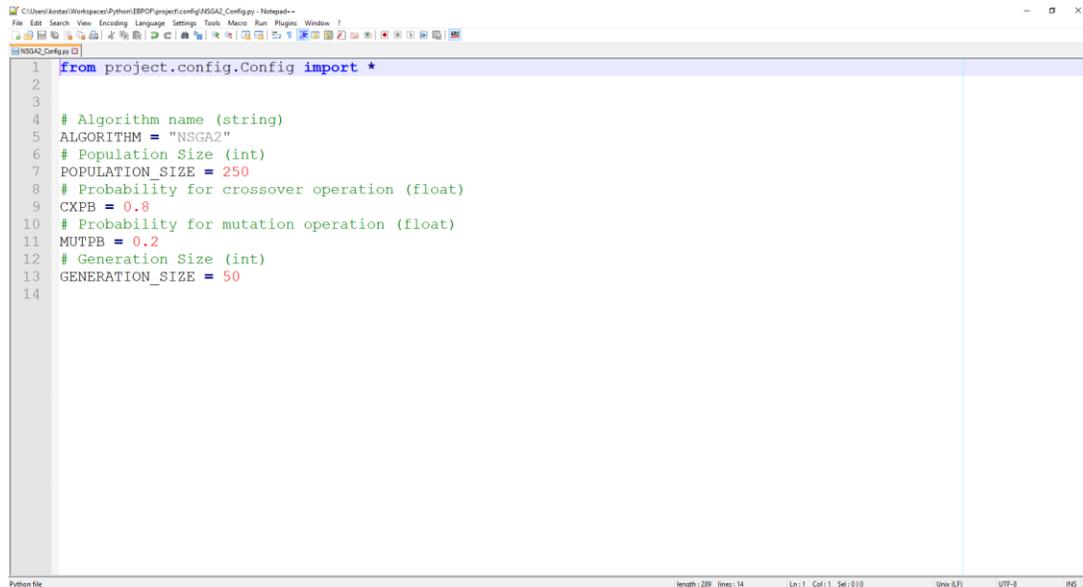


# Inputs of eBPO<sub>F</sub>

EMOAs employed in eBPOF: NSGA2, SPEA2 and DCD.

Each of them has distinctive features that enhance the optimization process.

For the selected EMOA, the user is responsible to provide a configuration file with a set of parameters that control the optimization operation.



```
1 from project.config.Config import *
2
3
4 # Algorithm name (string)
5 ALGORITHM = "NSGA2"
6 # Population Size (int)
7 POPULATION_SIZE = 250
8 # Probability for crossover operation (float)
9 CXPB = 0.8
10 # Probability for mutation operation (float)
11 MUTPB = 0.2
12 # Generation Size (int)
13 GENERATION_SIZE = 50
14
```

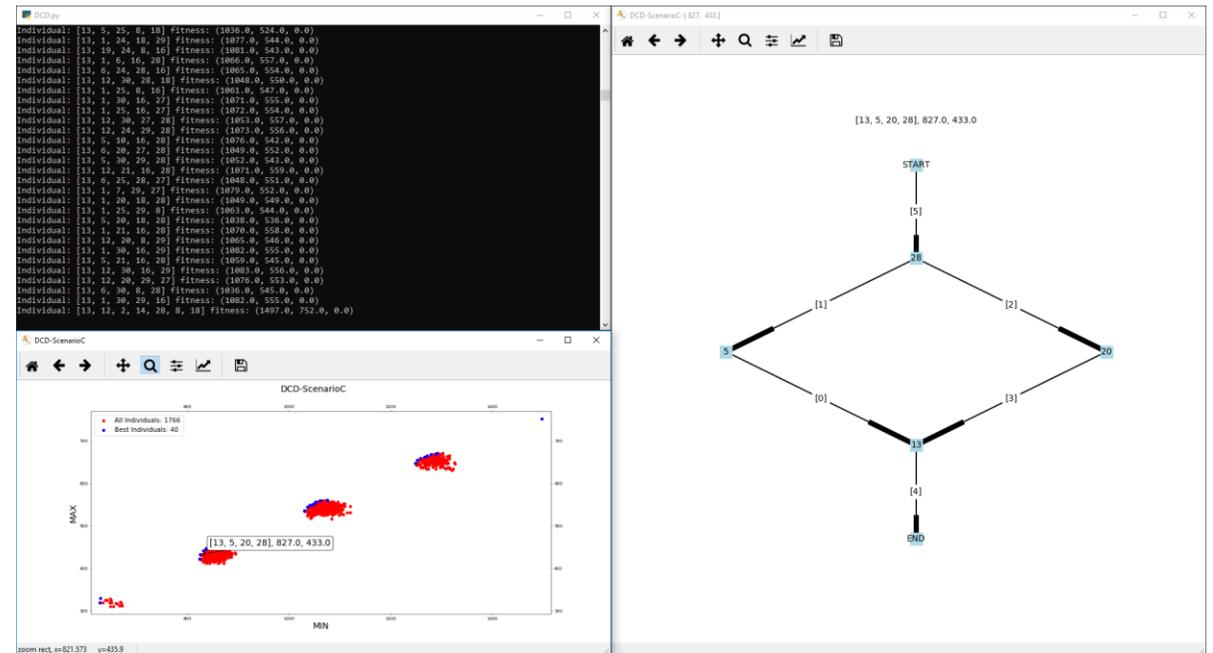
The screenshot shows a Python IDE window titled 'NSGA2\_Config.py'. The code defines a configuration class with the following parameters: ALGORITHM (string) set to 'NSGA2', POPULATION\_SIZE (int) set to 250, CXPB (float) set to 0.8, MUTPB (float) set to 0.2, and GENERATION\_SIZE (int) set to 50. The IDE interface includes a menu bar, a toolbar, and a status bar at the bottom showing file length (288), line count (14), and encoding (UTF-8).

# Outputs of eBPO<sub>F</sub>

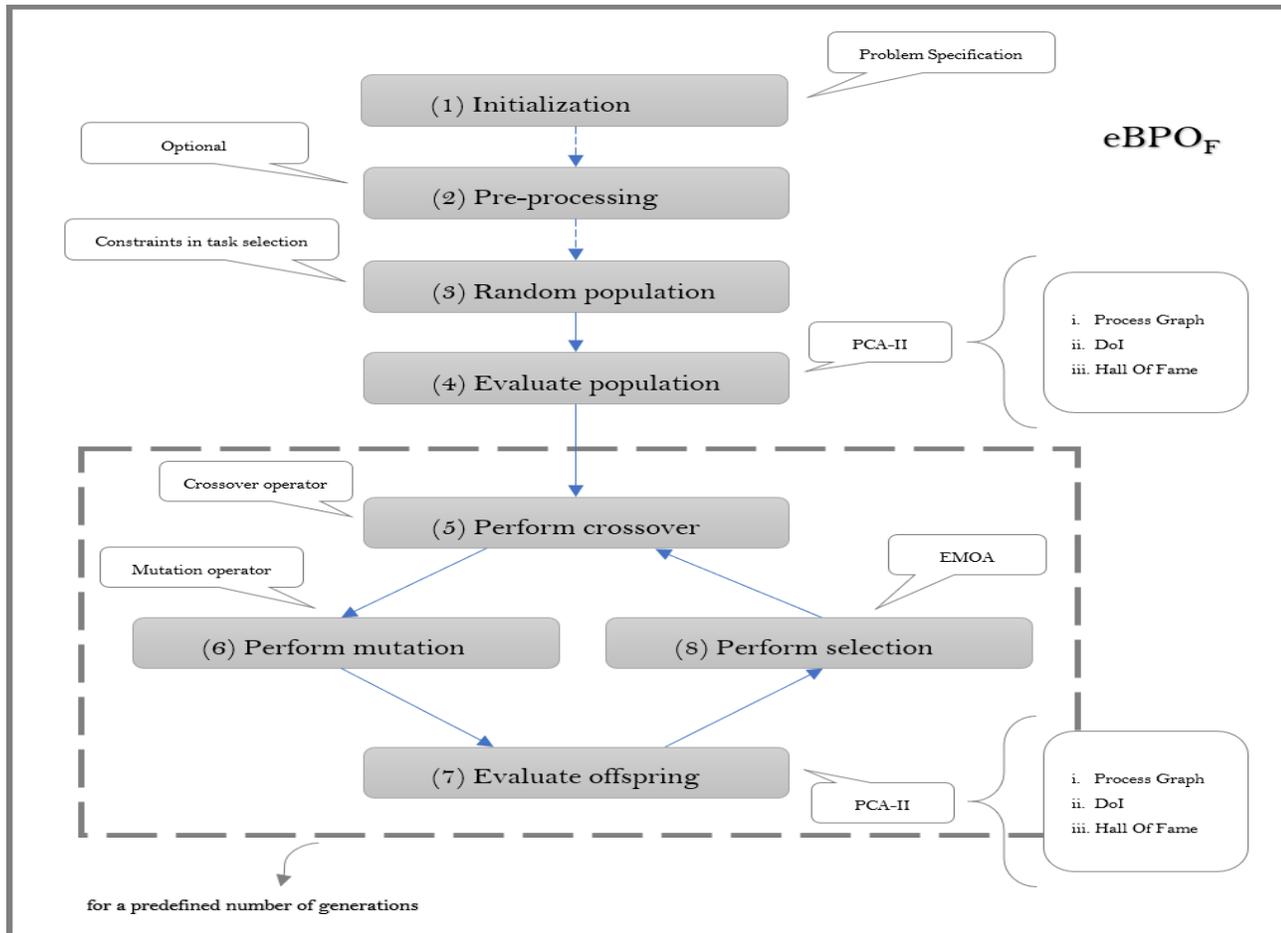
The proposed optimization framework generates a series of optimized business process designs using the inputs in conjunction with one of the evolutionary algorithms.

The outcome of the framework is a data structure called “Hall of Fame” which contains all feasible business process designs found during the execution of eBPO<sub>F</sub>.

From those designs, their pareto front is produced which comprises the optimized process designs found by eBPO<sub>F</sub>.

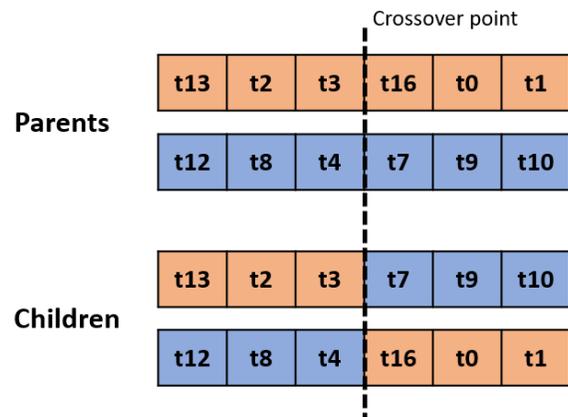


# Main steps of eBPO<sub>F</sub>

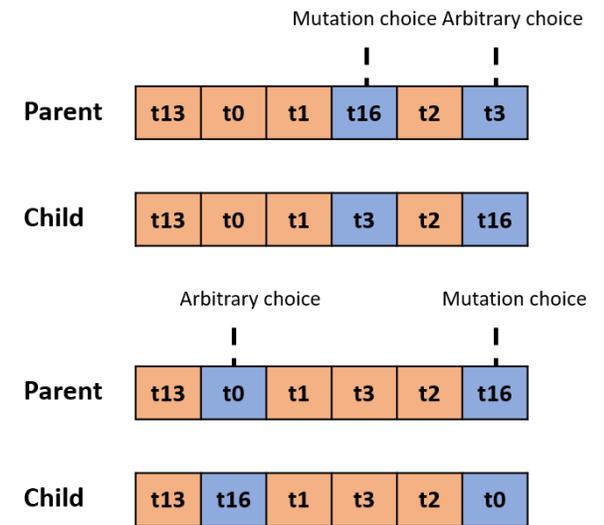
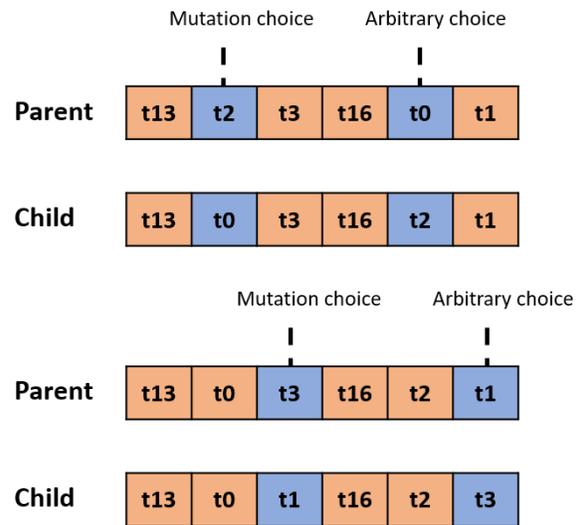


# Genetic Operators in eBPO<sub>F</sub>

## Crossover



## Mutation



# Testing Scenarios

Vergidis has developed three scenarios for the validation testing of BPO<sub>F</sub> by capturing the context of three different business processes of the service industry and creating the corresponding libraries of tasks by gathering relevant sub-services.

All scenarios are considered as bi-objective optimization problems and their problem type involves the minimization of the first objective and the maximization of the second one.

The first objective is the Service Delivery Price, (SDP), which specifies the amount of money that the service customer must pay to use the service.

The second objective is the Service Fulfilment Target, (SFT), which specifies the service provider's promise of effective and seamless delivery of the defined benefits to any authorized service customer requesting the service within the defined service times. It is expressed as the promised maximum number of successful individual service deliveries considering the total number of individual service deliveries.

# Configuration of Scenarios and EMOAs

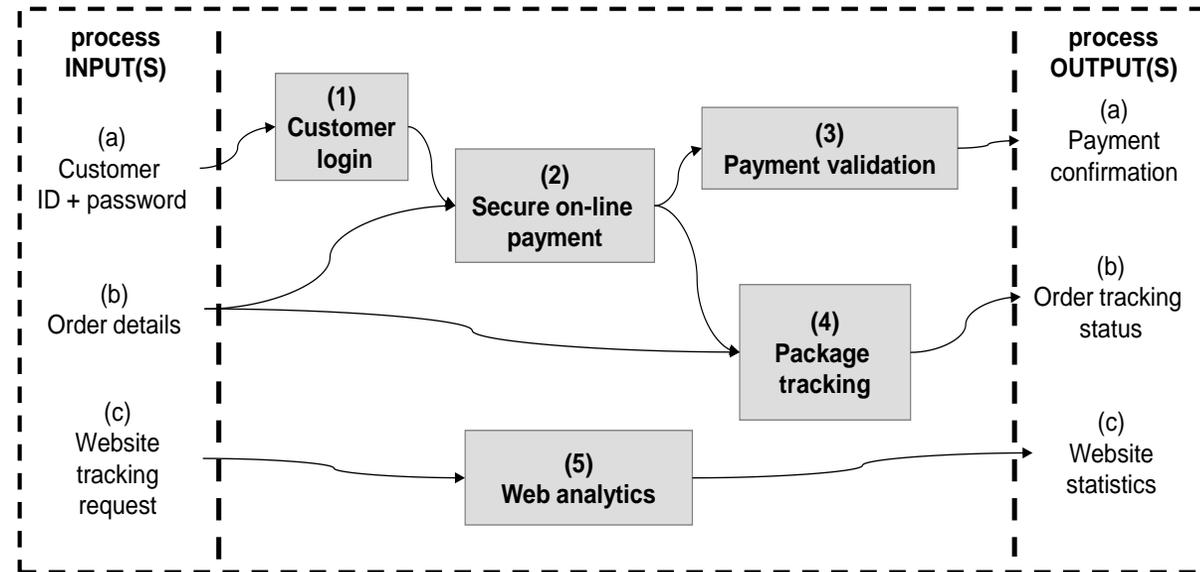
Each of the employed EMOAs has the same configuration for each scenario examined

Parameter	Value
Container size	10
Minimum size	0
Maximum size	10
Problem type	MIN-MAX

Parameter	NSGA2	SPEA2	DCD
Population size	100	100	100
Generation size	20	20	20
Crossover probability	0.8	0.8	0.8
Mutation probability	0.2	0.2	0.2

# Scenario A

Online order placement is an automated business process



# Pre-processing of Scenario A

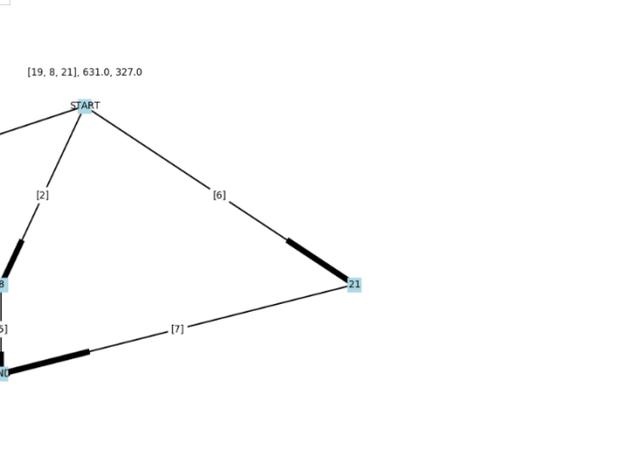
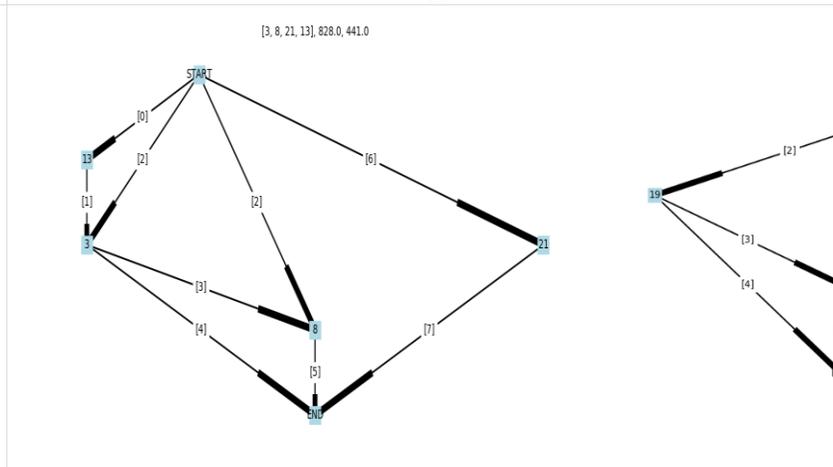
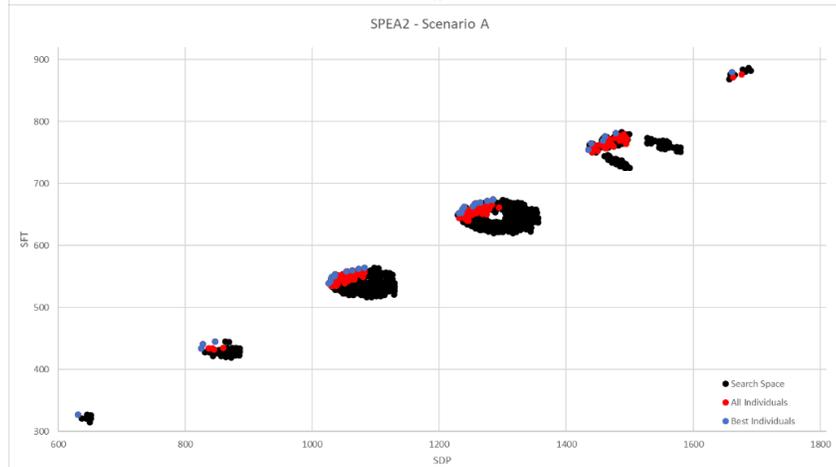
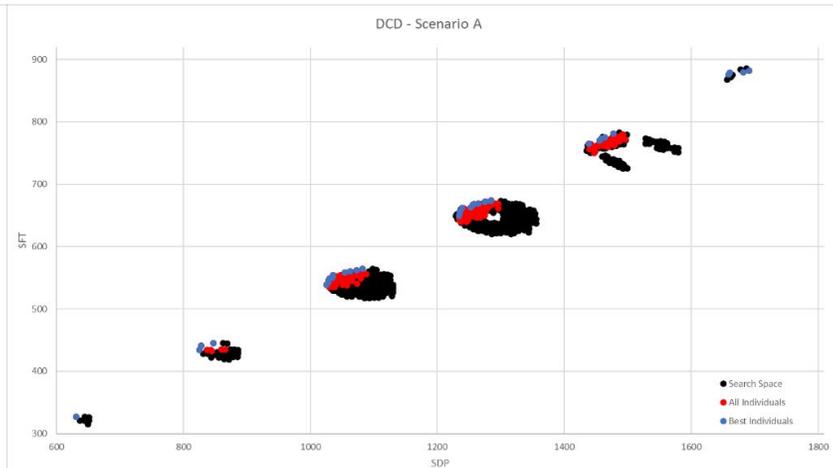
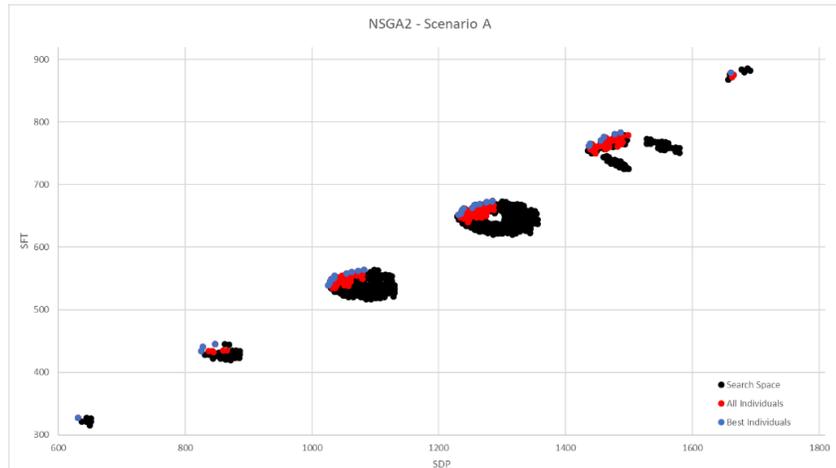
No.	Task Name	Input(s)	Output(s)	SDP	SFT	Valid
0	Achworks Soap (T\$\$ - Rico Pamplona)	1, 2	3	208	113	✓
1	BAX Global Tracking Service	2, 3	5	219	109	
2	CDYNE Death Index	1, 3	4	229	115	✓
3	Credit Card Processor	1, 2	3,4	202	109	✓
4	D&B Business Credit Quick Check	1, 3	4	203	108	✓
5	Drupal authentication	0	1	200	103	✓
6	ecomStats Web Analytics	6	7	218	112	
7	Entrust login	0	1	206	103	
8	FedEx Tracker	2, 3	5	211	109	✓
9	FedEx / UPS Package Tracking	2, 3	5	224	103	
10	FraudLabs Credit Card Fraud Detection	1, 3	4	220	113	✓
11	Google Analytics	6	7	218	107	
12	Google Checkout	1, 2	3	206	105	✓
13	GUID Generator	0	1	203	110	✓
14	Internet Payment Systems	1, 2	3	226	105	
15	LID login	0	1	222	114	✓
16	OpenID login	0	1	228	100	
17	Paypal online payment	1, 2	3	215	102	
18	Real Time Check Verification (T\$\$ - Rico Pamplona)	3	4	229	108	✓
19	Rich Payments NET	2	3, 4	208	105	✓
20	SAINTlogin users validation	0	1	219	105	
21	Servicetrack	6	7	212	113	✓
22	SmartPayments Payment	2	3	214	105	✓
23	Smartpayments CardValidator	3	4	206	107	✓
24	StrikeIron Global Address Verification	1, 3	4	201	105	✓
25	SXIP login	0	1	203	105	
26	Typekey authentication service	0	1	224	114	
27	UPS Tracking	2, 3	5	225	109	
28	VeriSign Payment	1, 2	3	230	103	

No.	Resource name	Process I/O
0	Customer account credentials	I
1	Customer account details	
2	Order details	I
3	Payment details	
4	Payment confirmation	O
5	Order tracking status	O
6	Website tracking request	I
7	Website statistics	O

Only 16 tasks have been left out of 29

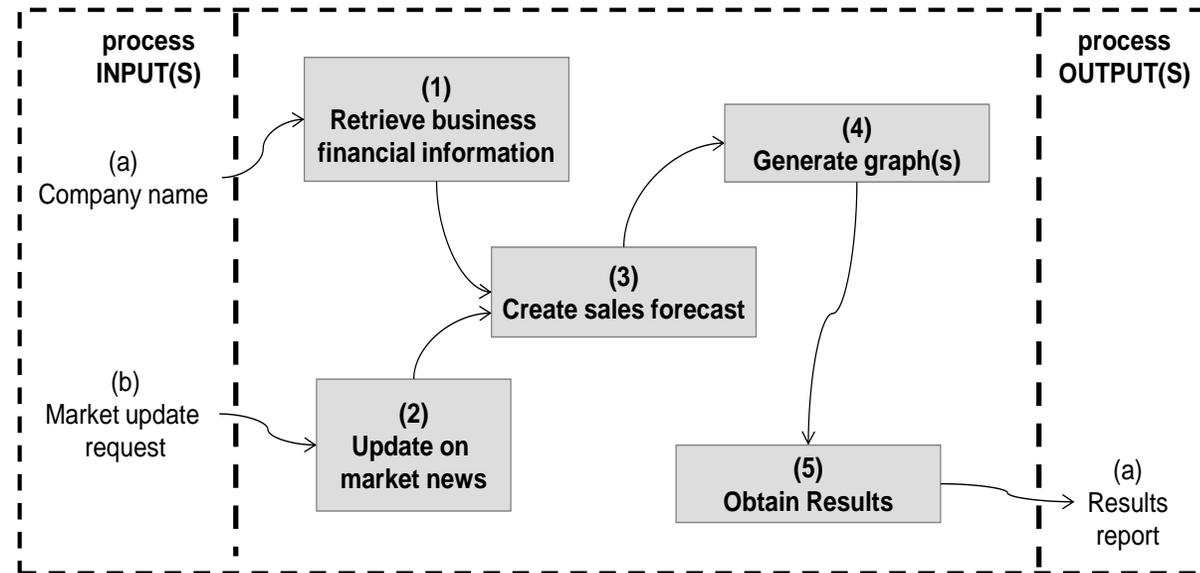
Decrease of the library size of 45%

# Validation testing of eBPO<sub>F</sub> with Scenario A



# Scenario B

Sales forecasting



# Pre-processing of Scenario B

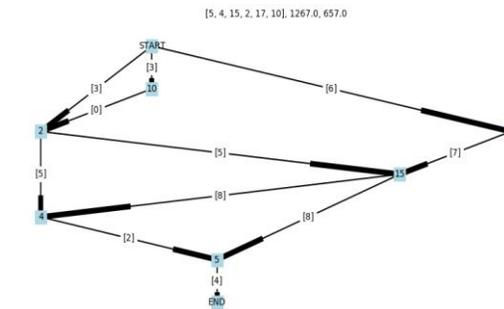
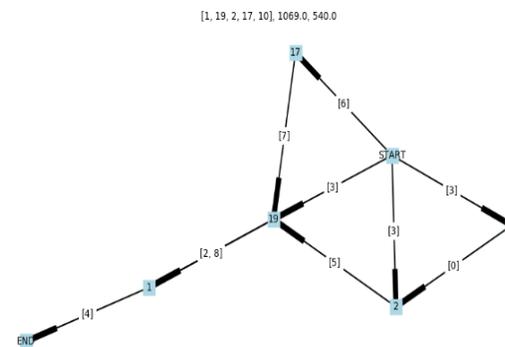
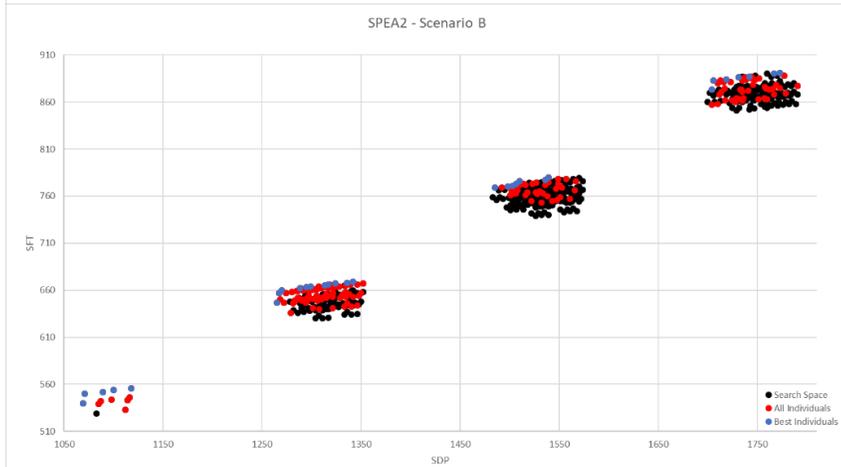
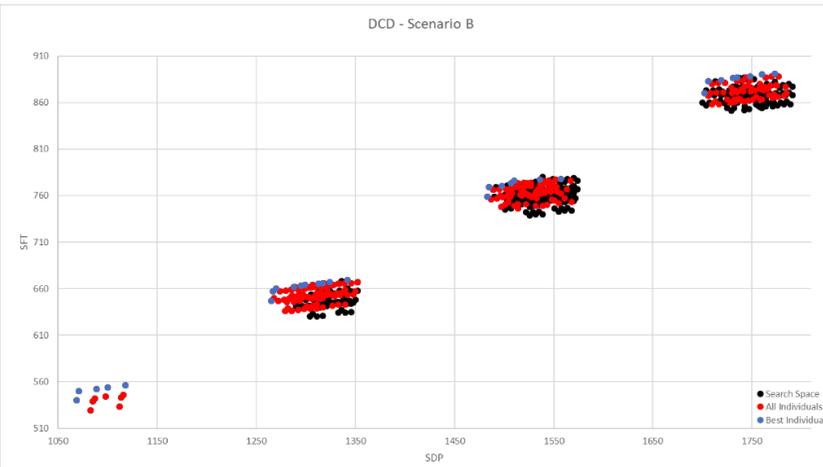
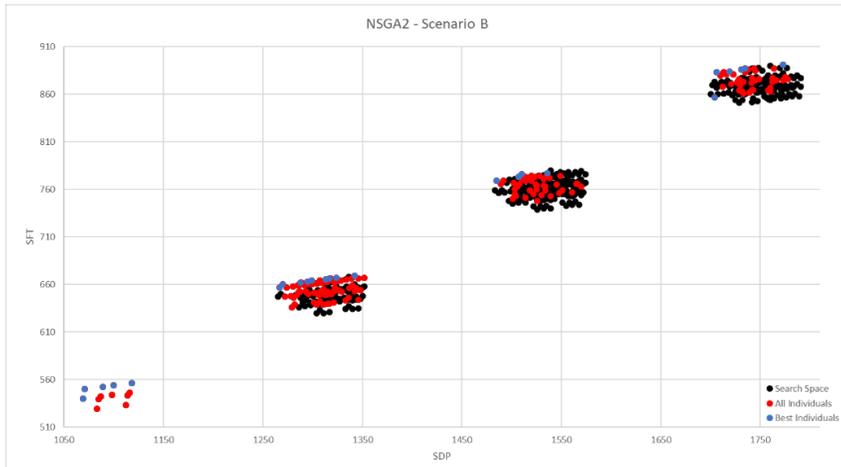
No.	Task Name	Input(s)	Output(s)	SDP	SFT	Valid
0	D&B Business Verification	3,1	0,5	206	103	
1	Fax.com	8,2	4	220	103	✓
2	Gale Group Business Information	3,0	5	223	106	✓
3	Gale Group Business Intelligence	3,1	0,5	229	113	
4	GraphMagic's Graph & Chart Web Service API	5,8	2	203	107	✓
5	interfax.net	8,2	4	222	113	✓
6	Lokad Business time-series forecasting and analysis	5,7	8	228	110	✓
7	Midnight Trader Financial News	6,3	7	217	101	✓
8	Strikelron Company Search	3	3,0	230	114	✓
9	Strikelron Get Business Prospect	3,1	0,5	205	110	
10	Strikelron Lookup Business	3	3,0	201	110	✓
11	Wall Street Horizon Real-Time Company Earnings	3,1	5	210	105	
12	Xignite Get Balance Sheet	3,1	5	216	112	
13	Xignite Get Chart Url	8	2	228	110	✓
14	Xignite Get Chart Url Preset	8	2	228	101	
15	Xignite Get Growth Probability	5,7	8	215	109	✓
16	Xignite Get Market News Headlines	6	7	221	114	✓
17	Xignite Get Market Summary	6	7	203	112	✓
18	Xignite Get Topic Chart	3,5,7	8	218	112	✓
19	Xignite Get Topic Data	3,5,7	8,2	222	109	✓

No.	Resource name	Process I/O
0	Business details	
1	Business query	
2	Chart / graph	
3	Company name	I
4	Fax (on-line)	O
5	Financial data	
6	Market update request	I
7	Recent market trends	
8	Time-series forecast	

Only 14 tasks have been left out of 20

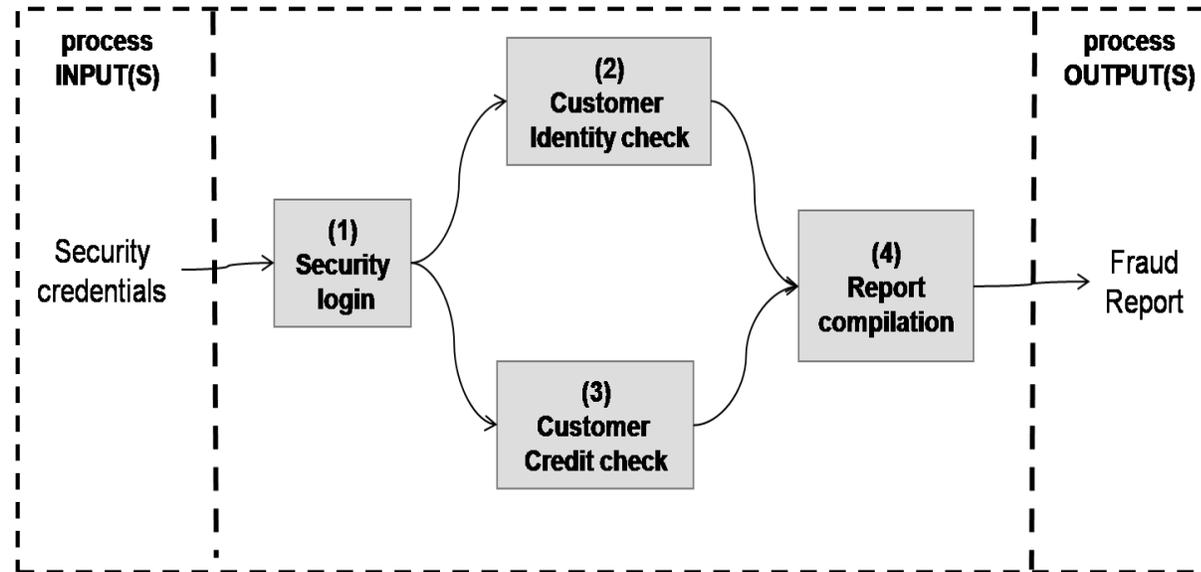
Decrease of the library size of 30%

# Validation testing of eBPO<sub>F</sub> with Scenario B



# Scenario C

Fraud investigation



# Pre-processing of Scenario C

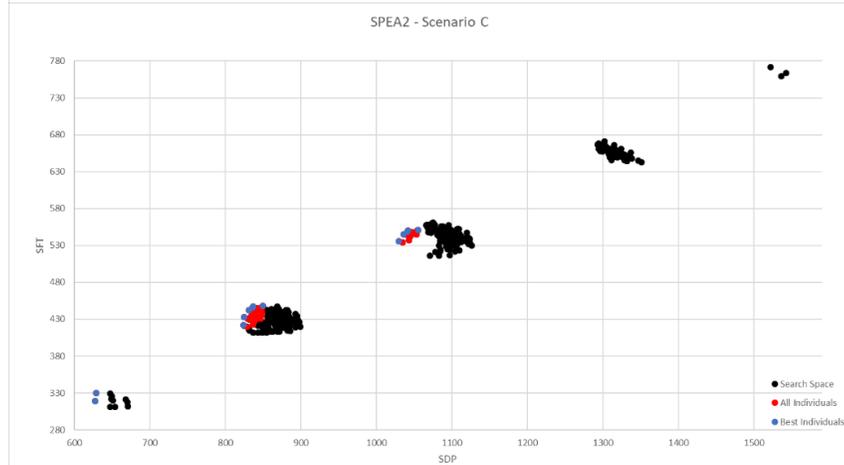
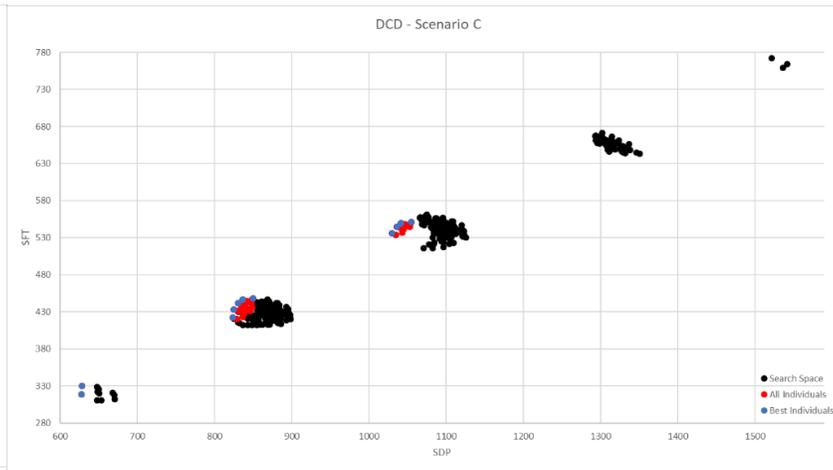
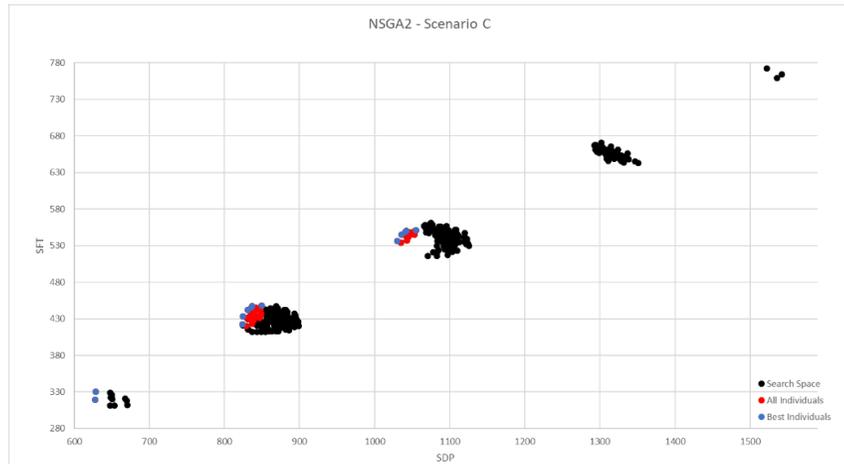
No.	Task Name	Input(s)	Output(s)	SDP	SFT	Valid
0	Address Doctor Global Address Verification	2	3	209	106	
1	cbarron bankValidate	1	0	212	114	✓
2	CDYNE Death Index	2, 1	3, 0	227	102	
3	CDYNE Email Verifier	2	3	210	105	
4	CDYNE Phone Verifier	2	3	212	109	
5	D&B Business Credit Quick Check	1	0	201	101	✓
6	D&B Business Verification	2, 1	3, 0	207	110	✓
7	Dimple Email Address Validator	2	3	208	112	
8	Drupal authentication	5	2, 1	205	103	✓
9	Dun & Bradstreet Business Credit Quick Check	1	0	215	109	
10	Dun & Bradstreet Business Verification	2	3	228	108	
11	Entrust login	5	2, 1	228	104	
12	FraudLabs Credit Card Fraud Detection	2, 1	0	213	115	✓
13	Google Docs	3, 0	4	216	106	✓
14	GUID Generator	5	2,1	219	109	
15	LID login	5	2,1	224	104	
16	OpenID login	5	2, 1	225	113	
17	Real Time Check Verification (T\$\$ - Rico Pamplona)	1	0	211	109	✓
18	SAINTlogin users validation	5	2, 1	211	103	
19	Smartpayments CardValidator	1	0	224	110	
20	Strikelron 24-hour Accurate Residential Lookup	2	3	204	112	
21	Strikelron 24-hour Accurate Reverse Phone Lookup	2	3	211	111	
22	Strikelron Email Verification	2	3	215	113	✓
23	Strikelron Gender Determination	2	3	215	102	
24	Strikelron Global Address Verification	2	3	211	111	
25	Strikelron Reverse Phone Residential Intel	2	3	203	111	
26	Strikelron Reverse Residential Lookup	2	3	222	107	
27	SXIP login	5	2, 1	216	110	
28	Typekey authentication service	5	2, 1	206	114	✓
29	Web Services Security Monitor	5	2, 1	227	110	
30	webba E-Mail validator	2	3	202	112	✓

No.	Resource name	Process I/O
0	Credit assessment	
1	Customer Credit details	
2	Customer ID details	
3	ID verification outcome	
4	Risk Assessment Report	O
5	Security login credentials	I

Only 10 tasks are still valid out of 31

Decrease of the library size of 68%

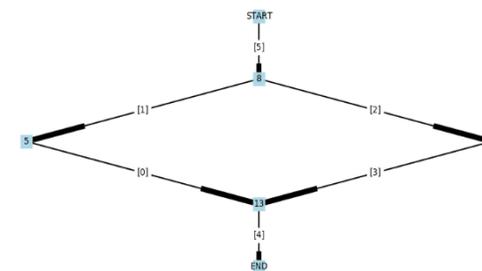
# Validation testing of eBPO<sub>F</sub> with Scenario C



[13, 6, 8], 628.0, 319.0



[13, 5, 30, 8], 824.0, 422.0





Any questions?

Thank you for attention!

