

Application of multiple population genetic algorithm in optimizing business process

Nadir Mahammed^a, Mahmoud Fahsi^b and Souad Bennabi^d

^aLabRI-SBA Laboratory, Ecole Supérieure en Informatique Sidi Bel Abbès, P.O 73, post office El Wiam Sidi Bel Abbès 22016, Algeria

^bDjillali Liabes University, P.O 89 Sidi Bel Abbès 22000, Algeria

^dHassiba BenBouali University, Ouled Fares Chlef 02180, Algeria

Abstract

In a highly competitive environment, enterprises success depends on the effectiveness of their business processes, which leads to the search of a continuous improvement in the time. This kind of improvement is called business process optimization. Yet, two major challenges often prevent processes optimization. First, the skills of the analysts to choose the right process among a number of propositions. Second, the techniques applied to generate and evaluate solutions during optimization process are poor and do not include all relevant data. Our Evolutionary Business Process Optimization approach addresses these challenges through a well-defined mathematical representation and a novel evolutionary algorithm as optimization facilities. In this paper, we focus to use of a formalized process optimization approach for generating and improving business process designs

Keywords

Multi-criteria optimization, genetic algorithm, business process, multiple population,

1. Introduction

A business process (BP) can be defined as a set of tasks that when properly connected perform a business operation, e.g. a service or a product with benefit for the enterprise [1]. The main elements of a BP are the resources, the tasks and their attributes. Each BP design is evaluated with their tasks' attributes (discrete values that describe a task, such as cost and duration). The present definition does not include the notion of sub-process.



Coello Coello in [2] defines a multi-criteria optimization as finding a set of best possible solutions where not one but several in conflict objectives are involved. Such problem can be solved using either mathematical programming techniques or metaheuristics [3] Coello Coello in general; in particular, evolutionary computing is used. By their simplicity of implementation, genetic algorithms (GA) have been used in several studies on multi-criteria optimization of BPs [4].

This article explores a way to build satisfactory BP designs in an evolutionary multi-criteria optimization problem. What authors mean by a BP design (as defined by Adamo in [5]) gener-

IAM'20: Third conference on informatics and applied mathematics, 21–22 October 2020, Guelma, ALGERIA

✉ n.mahammed@esi-sba.dz (N. Mahammed); mfahci@univ-sba.dz (M. Fahsi); s.bennabi@univ-chlef.dz (S. Bennabi)

 0000-0001-7865-5937 (N. Mahammed)

 © 2020 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
 CEUR Workshop Proceedings (CEUR-WS.org)

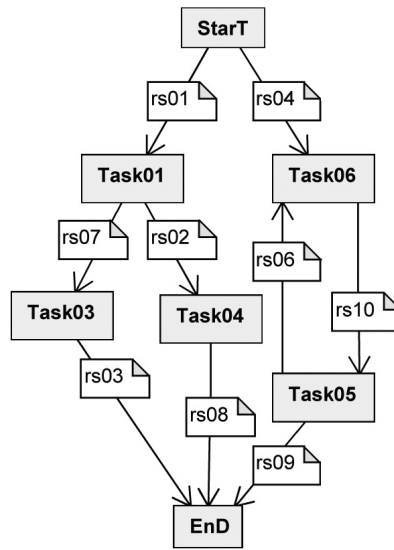


Figure 1: Business Process Design Representation

ated from a BP model is a BP where each composing task is described by one or more existing functional entity as web service and not a general description as a use case. A satisfactory, feasible or functional BP design is a BP which consumes all the input resources and produces all output resources as defined by the original BP model.

The result is an approach dealing with an evolutionary multi-criteria business process optimization (BPO) issue by minimizing cost and duration. The approach utilizes a multiple population GA adapted for the stated problem. The rest of the paper is organized as follows. Section 2 presents a state of the art on BPO with evolutionary algorithms (EA). Section 3 presents the optimization approach with the mathematical representation of BP designs and introduces properly the proposed multi-population GA. Experimental results are presented in section 4. Finally, section 5 summarizes the paper with conclusion and directions for future work.

2. Related work

The work of Hofacker [6] is the first to quote. The authors worked on how to compose each BP design in a correct way to guarantee their high performance by focusing on the resources' allocation to activities. Vergidis in [7] and [8] presented major work in BPO using EAs. They proposed an effective approach for automated multi-objective optimization of BP designs with multiples EAs leading to promising results. Vergidis in [9] came back proposing an interesting improvement of their optimization approach. The work is summed by considering each activity composing a potential BP design as a Web service with its features. Cited work have approached the problem of multi-objective optimization of BPs regarding to up to two criteria to optimize. But, Mahammed in [10] worked on a multi-objective combinatorial optimization problem up to three criteria. The authors proposed an approach that combines a novel geo-

metric Fitness function with an improved NSGA-II. [12] improved the results of [9] work.

Georgoulakos in [11] resumed [9] proposal and added a pre-processing stage to categorize Web services for the working optimization framework. Thereafter, the work of Tsakalidis [12] summarizes Vergidis' work on BPO with evolutionary computing. In Si [13], the BPO is discussed in regards to resources allocation. A GA is used with color Petri nets to reach desired results. The authors of [4] worked on the improvement on BPO by using a proposed selection operator independent from Fitness function calculation, leading to improve [11]] results in time execution and number of appropriate solutions. The authors in Comuzzi work [14] looked to the BPO by proposing a technique to implement process navigation of BP models as a restricted class of directed hypergraphs using an EA, the ant-colony optimization algorithm. Kurniasih in [15] proposed to study the query processing on data storage to maintain the performance of BP of a system. They used a modified memetic algorithm based on GA combined with local search technique and a tabu search technique on the crossover operation. In [16], Shajahan searched to optimize a specific BP, the product development process using a design structure matrix and a particle swarm optimization technique for the design and sequence of activities composing the BP. Rekik in [17] proposed to optimize BP in outsourcing by focusing on the selection of BP activities and the most suitable infrastructure as a service in the cloud life cycle. They proposed a system that used NSGA-II. Neira-Tovar [18] presented a study enabling to identify the pertinent elements of a methodology that allows the development of a model to optimize BPs. Deshmukh [19] explored how to generate several competing BP models from event logs using EAs in multi-objective optimization for process mining. Another interesting work, Li [20] aimed to optimize BP by proposing an approach based on a GA to improve the efficient of selection and composition of resource-service chain in a distributed cloud manufacturing problem.

This article describes authors' investigation and experimentation on the benefit from using a multiple population GA in BPO. The idea to propose the use of a multiple population within a GA comes from authors previous successful work which implied using a modified GA but still with a unique population. The authors are working to demonstrate that applying a multiple population approach in such issue can lead to explore more search space and in a shorter execution duration, at the same time.

3. Optimization approach

3.1. Mathematical Representation

The development of a functional and efficient optimization approach needs to overcome a number of variant challenges:

1. A simple representation technique able to capture the main features of a solution and easily used by optimization methods (e.g. GA).
2. An algorithmic approach for the composition of new solutions and evaluating them in a good manner.

During the optimization process, the authors work on the appended value by using a multiple

Table 1
Business Process Design Example

	StarT	T ₀₁	T ₀₂	T ₀₃	T ₀₄	T ₀₅	T ₀₆	EnD
StarT		r ₀₁						
T ₀₁				r ₀₇	r ₀₂			
T ₀₂								
T ₀₃								r ₀₃
T ₀₄								r ₀₈
T ₀₅							r ₀₆	r ₀₉
T ₀₆						r ₁₀		
EnD								

population GA with the aim of reducing execution constraints such as calculation duration, and improve execution results.

A solution is defined by authors as a BP design has a mathematical representation (Table 1) which results to a graphical representation: a Task Task Matrix (*TTM*) (see figure 1) and its calculated attributes values. In order to represent a BP design, two nodes are added *StarT* and *EnD*, which represent the beginning and the end of a design, respectively. Each design has tasks (T_i) that consumes and produces one or more resources (rs_i).

- If $TTM_{ij} = rs_k$ then $T_i \cap T_j = rs_k$
- If $TTM_{ij} = \{\}$ then $T_i \cap T_j = \{\}$

3.2. Multiple Population Genetic Algorithm

Non-dominated sorting genetic algorithm II or NSGA-II is an EA that helps to solve this category of problems. It conducts the non-dominated sorting of the initial population, then applies genetic operators to obtain a brand-new population, merges the parent population with the resulting population, and conducts a non-dominated sorting to obtain a nondominated solutions front.

In former version of NSGA-II, a single population is maintained during the algorithm execution. A consequence of such choice is that population diversity may be poor as many local optima in some multi-objective optimization issues. Using a multiple population method with an EA helps to accelerate the convergence and because it uses multiple sub-populations, it accelerates individuals' migration too, compared to a simple and unique population, which leads, to improve the results quantity.

Tanese [21] and Pettey [22] were the first to add a multiple population aspect with a GA based on the parallel computing paradigm. Chen in [23] describes two manners in which individuals are selected to migrate:

- Uniform selection chooses individuals and replaces them in a sub-population with individuals randomly.

- Selection based on Fitness picks individuals with high Fitness level and replaces them in a sub-population uniformly at random.

Chen in [23] proposed three migration manners between sub-populations of those being selected individuals.

- Adjacent migration: When individuals are allowed to transfer between adjacent sub-populations unidirectional.
- Neighborhood migration: When individuals are allowed to transfer between nearest sub-populations in either direction.
- Free migration: When individuals are allowed to transfer from any sub-population to another.

The authors have developed a novel multiple population GA based on NSGA-II which called MP-NSGA-II. The core functioning in figure 2:

1. An initial population of n individuals is randomly generated.
2. The population is sorted in m sub-population based on the crowding distant of NSGA-II.
3. The population is decomposed in m sub-population based on the sorting result.
4. NSGA-II is applied separately on each population.
5. After a defined number of iterations all sub-populations share their information (Figure 2):
 - Best individuals of each sub-population are chosen to migrate.
 - A new main population is formed by the combination of best individuals (Fitness-based) of each sub-population using a migration rate.
6. The n best individuals are kept to form the new parent population.
7. Steps 2 to 6 are repeated for a number of iterations.

3.3. Approach Description

The proposed approach generates a set of optimized BP designs from an initial BP model. Its inputs and its outputs resources primarily define the model. Figure 3 outlines the steps making up the approach.

1. Perform MP-NSGA-II. The approach begins with the production of a set of populations of BP designs randomly. Which happens once during the approach execution. The population is not assessed until the evaluation stage (step 4). During the future iterations, MP-NSGA-II is applied rigorously.
2. Create mathematical representation. After the generation of a population of n BP designs, each individual is represented by its *TTM* (Table 1). Each matrix represents the relationship between the tasks composing a design using resources.

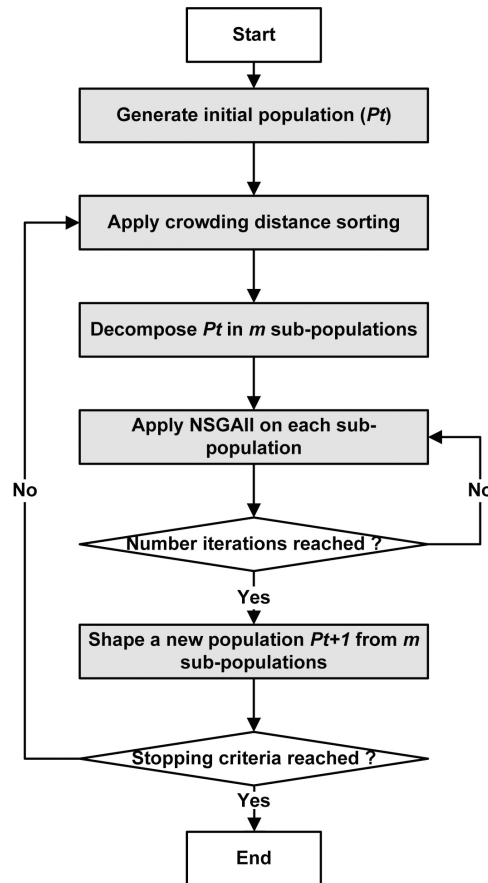


Figure 2: MP-NSGA-II Flowchart

3. Apply and verify a set of restraints. During the approach course, a set of requirements needs to be examined regarding to the input and output resources, used and produced by each solution, respectively. Then, checks a set of constraints regarding the design itself.
4. The evaluation consists of calculating the attributes of each individual created and on which different constraints have been applied. The attributes of a BP design are calculated based on the tasks' attributes composing it. Subsequently, these attributes are the objective values used to calculate the Fitness function of each solution which used later by MP-NSGA-II.
5. Apply Steps 1 to 4 are repeated for a number of iterations defines by the experimenters. It would be advisable to specify that the execution of the first step differs after the first iteration.
 - MP-NSGA-II generates the first population randomly from scratch during the first iteration.
 - MP-NSGA-II generates the new population (always) randomly but from the offspring of the first iteration results, during all of the following iterations.

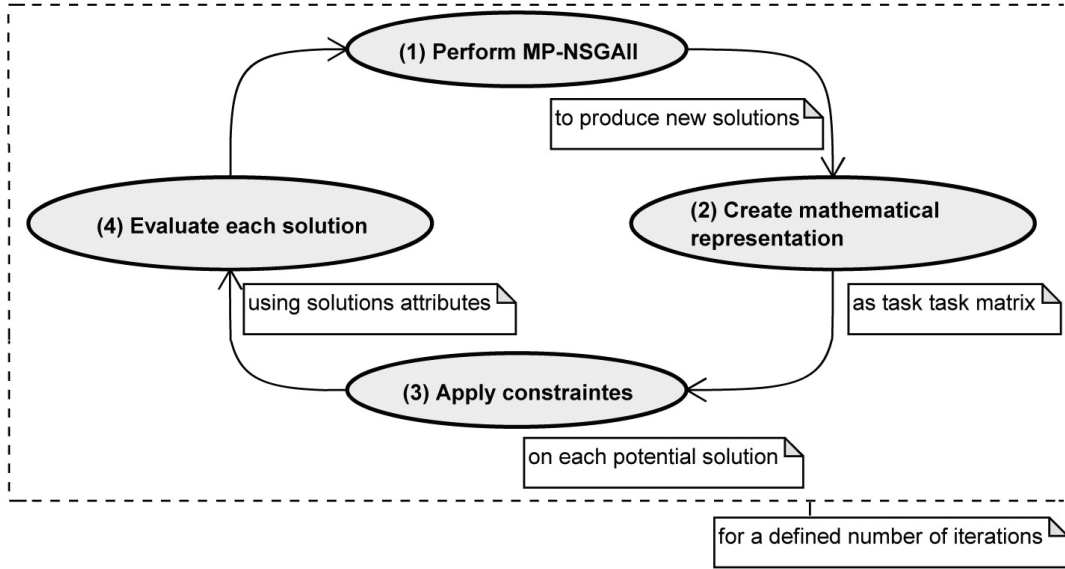


Figure 3: Optimization Approach Steps

4. Experimentation

The proposed optimization approach aim is to generate a set of optimized feasible BP designs. A feasible solution is a BP design which:

1. Consumes all input resources of the original BP model,
2. Produces all output resources of the original BP model.
3. With limited length.
4. It leads to define the features and summarizes what is expected from the approach as:
 - The amount of feasible non-dominated solutions.
 - The solutions with different sizes and different solutions with same size i.e. solutions variety

4.1. Design and Measures

The present work focuses on a combinatorial multi-criteria BPO and seeks to minimize two criteria. These two criteria' units of measure are not specified. As example, criteria can be assumed as cost and duration, and the aim is to minimize them on a design as studied in [24].

The authors assumed that tasks composing a solution are in sequential order. So, the calculation of a BP design attributes (objective values) for the tasks is shown in equation (1):

$$bpd_i(a_j) = \sum_{k=1}^{n_d} a_{jk} \quad (1)$$

- bpd_i : BP design.

Table 2
Optimization Approach Parameters

Parameters	Values
Initial population size	1000
Binary tournament selection	-
One-point crossover	0.9
Bitwiser mutation	$1/k$
Number sub-populations	5
Number individuals per sub-population	200
Migration period	30 generations
Migration rate	0.2
Insertion rate	0.8

- n_d : Tasks' number composing bpd_i .
- a_j : Objective value.
- a_{jk} : Value of a_j for a task k .

The Fitness function used by MP-NSGA-II where k the number of criteria to optimize is in equation (2):

$$f(bpd_i) = \sum_{j=1}^k bpd_i(a_j) \quad (2)$$

To explore the influence of the multiple sub-population feature on the performance of MP-NSGAII, it seems necessary to know how to measure it. Therefore, as described in [25], an average optimal value (*AOV*) and a success rate (*SR*) are used in this study to measure the performance of MP-NSGA-II. For a given population size, the algorithm runs several times. Table 2 summarizes the optimization approach parameters. It itemizes also the genetic operators features used within MP-NSGA-II.

4.2. Materials

First, the impact of the sub-population size, and the sub-population number, affect the performance of MP-NSGA-II is studied. The results are shown in figure 4 for the sub-population number experiments and figure 5 for the sub-population size experiments.

The experiment results of the optimization approach are presented then analyzed using five (05) different scenarios borrowed from [9] and [11]. The scenarios gather a different tasks library size (set of tasks used to generate solutions).

Tables ?? and ?? show computational results where all tests were performed using the settings summarized in Table 2. For the purpose of comparison, the authors conducted experiments on two axes, with one hundred (100) iterations each:

- Diversifying used GAs: Simple genetic algorithm (SGA), NSGA-II, MP-SGA and MP-NSGA-II.

Table 3
Test Scenarios Parameters

Scenarios	Resources number	Tasks library size
A	20	9
B	30	11
C	50	30
D	500	30
E	1000	30

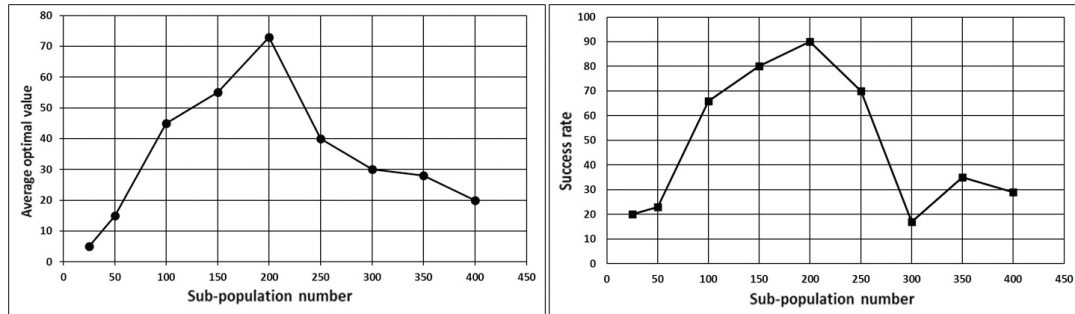


Figure 4: Effect of Sub-population Size on MP-NSGA-II

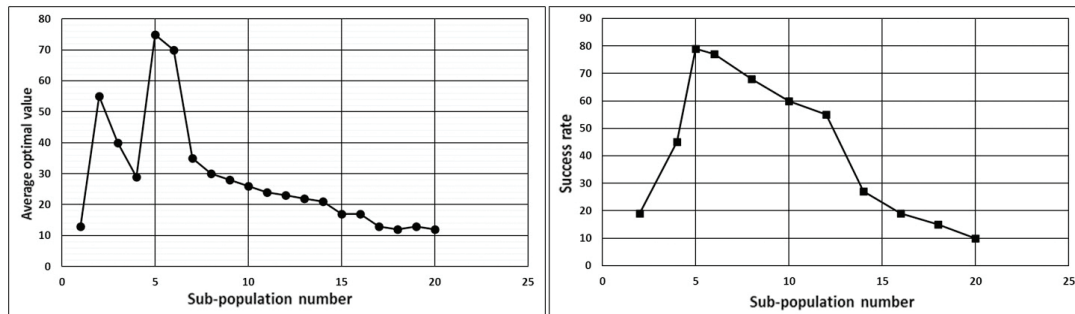


Figure 5: Effect of Sub-population Number on MP-NSGA-II

- Diversifying migration process: Adjacent migration, neighborhood migration and free migration.

4.3. Results and Discussion

The influence of the sub-populations size on the performance of MP-NSGA-II is studied (likewise for the sub-population number). The size is changed from 0 to 450 with a step size of 50. Figure 4 presents the experiments results. The algorithm runs 50 times, and the average of the values of these runs are used to measure the performance of MP-NSGAI.

In figure 4, the X-axis, Y-axis (left), and Y-axis (right) represent sub-population size (S), AOV,

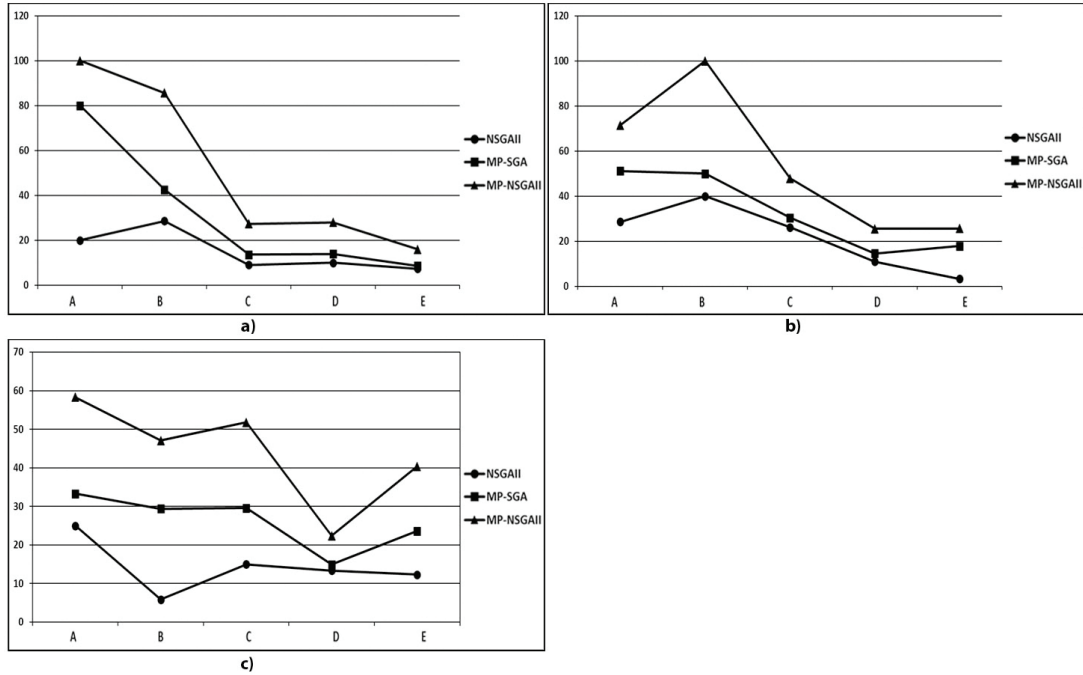


Figure 6: Solutions increase rate (%): a) Adjacent migration process. b) Neighborhood migration process. c) Free migration process

and SR , respectively. For $S < 50$, the performance of the algorithm is extremely poor ($AOV < 10$ and $SR = 20$). For increasing S , the performance of MP-NSGA-II improved rapidly. For $S = 200$, MP-NSGA-II shows an improved performance with an AOV of 73, and SR is 90%. At this level, MP-NSGA-II performs best. However, as S continues to increase, the performance of MP-NSGA-II slowly decreases. Therefore, it can be concluded that, the performance of proposed algorithm improves rapidly with increasing S , and then decreases slowly. To obtain an improved approach, the value of the size of sub-populations cannot be too small, specifically, a value around 200 is a good choice.

Second, the influence of the sub-population number on the performance of MP-NSGA-II is studied. The number is changed from 0 to 25 with a step size of 5. Figure 5 presents the obtained results. In figure 5, the X-axis, Y-axis (left), and Y-axis (right) represent number of sub-populations, AOV , and SR , respectively. For $N < 20$, the performance of MP-NSGA-II is poor and AOV is less than 20, and SR is 20%. The performance of MP-NSGA-II increases slowly with increasing of the number of sub-populations (N). For $N = 5$, MP-NSGA-II performs the best, $AOV = 75$, and $SR = 79\%$. As N continues to increase, the performance of MP-NSGA-II rapidly decreases. Therefore, it can be concluded that, the performance of MP-NSGA-II first improves slowly and then decreases rapidly with increasing N . To obtain an improved optimization approach, the value of sub-populations should be a value around 5, for the present issue. These values ($S = 200$, $N = 5$) is used later on the forthcoming experiments as shown in the approach parameters (Table 2).

The number of non-dominated solutions by scenario with the optimization approach using

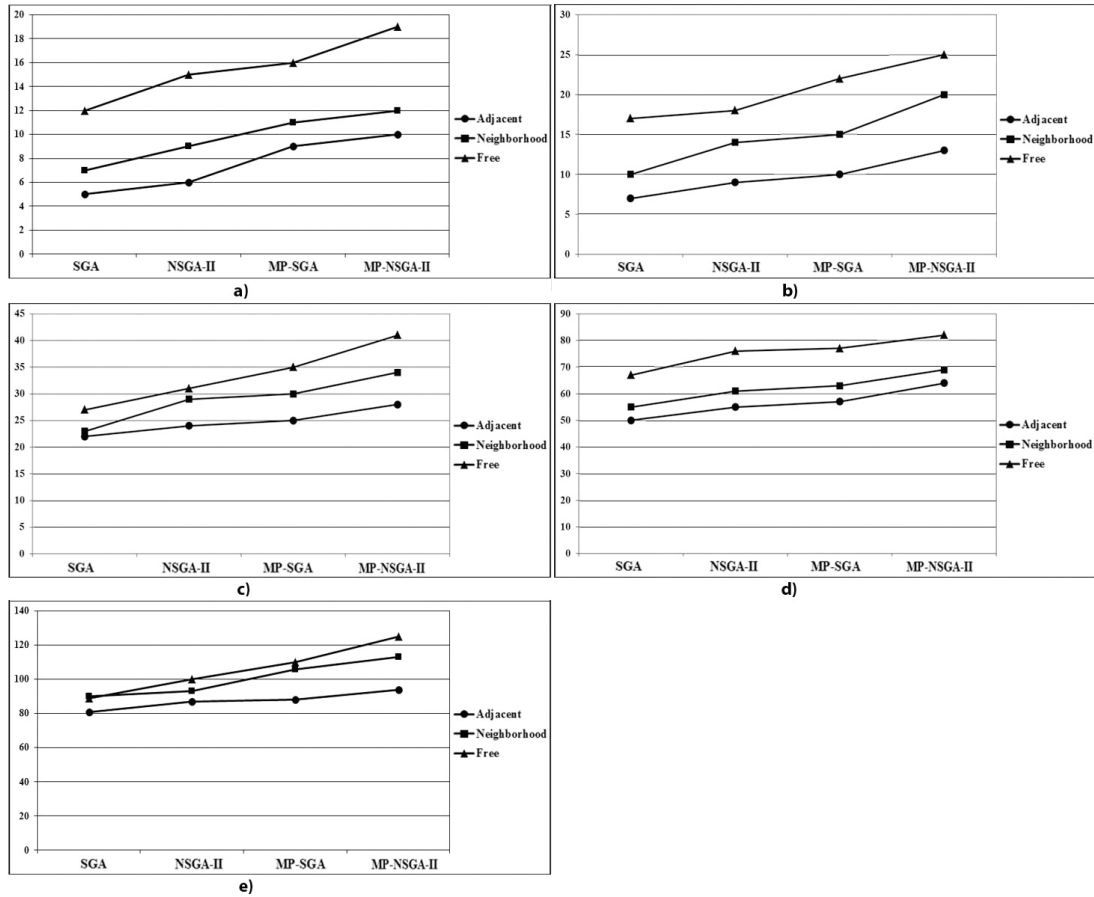


Figure 7: Solutions increase rate (%): a) Scenario A. b) Scenario B. c) Scenario C. d) Scenario D. e) Scenario E

multiple population GAs with different migration processes is summarized in figure 6. The amount of solutions obtained differs between the different GAs can be observed evidently. Among the four tested GAs, MP-NSGA-II outperforms on each scenario and with every migration process applied. Even if the values are close (by migration process) for each scenario, which can be explained by the discrete nature of values used in different scenarios (tasks library and available resources). But, if the comparison is done by focusing on the GA instead of different migration process, the obtained results differ significantly.

Figure 6 presents the increase rate (%) of number of solutions generated compared to a simple GA. Figure 7 depicts the efficiency of the proposed multiple population GA for different scenarios. It summarizes the average results classified by scenario. It clearly highlight and confirms that the different versions of the approach with multiple population GAs obtain the best overall results compared with a simple GA or NSGA-II. It is clear that for scenarios sample in the experiments, the performance of MP-NSGA-II is significantly better than the single-population GA not only in the number of results, the execution time but also in the convergent speed of searching. In addition, since the multi-population GA let the sub-populations evolve

separately, it might be able to explore more extreme solutions.

5. Conclusion

A GA based on multi-population (MP-NSGA-II) is presented in this study for solving a multi-objective BP designs problem. The algorithm can be summarized as follow: First, sub-populations are created from a unique initial population, using a crowding distance sorting. Then, in order to prevent the premature convergence of all individuals into a local optimal, NSGA-II is applied separately on each sub-population. Finally, sub-populations are regrouped (by migration) as a single population. To do so, each individual in the final population is chosen based on a non-dominated sorting randomly which may help to explore for more solution spaces next iterations.

So, this paper explores the idea of using MP-NSGA-II in a multiple objective problem of BPO. This work concretely aims at giving a tool capable to enhance the generated BP designs in quantity and diversity. Further investigations and experimentation would be welcomed to improve the proposed approach and its EA. In the future, tests with MP-NSGA-II can be further extended to a larger number of tasks to work with. Working with more optimization criteria is the next step to follow. It would be interesting to manipulate the number of sub-populations and population size within the chosen GA.

References

- [1] N. Mahammed, S. M. Benslimane, A. Ouldkradda, M. Fahsi, Evolutionary business process optimization using a multiple-criteria decision analysis method, in: 2018 Inter. Conf. on Computer, Information and Telecommunication Systems (CITS), IEEE, 2018, pp. 1–5.
- [2] C. Coello Coello, S. De Computación, C. Zacatenco, Twenty years of evolutionary multi-objective optimization: A historical view of the field, *IEEE computational intelligence magazine* 1 (2006) 28–36.
- [3] C. A. C. Coello Coello, G. B. Lamont, D. A. Van Veldhuizen, et al., *Evolutionary algorithms for solving multi-objective problems*, volume 5, Springer, 2007.
- [4] N. Mahammed, S. M. Benslimane, N. Hamdani, Evolutionary multi-objective optimization of business process designs with ma-nsgaii, in: IFIP Inter. Conf. on Computational Intelligence and Its Applications, Springer, 2018, pp. 341–351.
- [5] G. Adamo, C. Ghidini, C. Di Francescomarino, What's my process model composed of? a systematic literature review of meta-models in bpm, *arXiv preprint arXiv:1910.05564* (2019).
- [6] I. Hofacker, R. Vetschera, Algorithmical approaches to business process design, *Computers & Operations Research* 28 (2001) 1253–1275.
- [7] K. Vergidis, A. Tiwari, Business process design and attribute optimization within an evolutionary framework, in: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 668–675.
- [8] K. Vergidis, A. Tiwari, B. Majeed, Business process improvement using multi-objective optimisation, *BT Technology Journal* 24 (2006) 229–235.

- [9] K. Vergidis, C. Turner, A. Alechnovic, A. Tiwari, An automated optimisation framework for the development of re-configurable business processes: a web services approach, *Inter. J. of Computer Integrated Manufacturing* 28 (2015) 41–58.
- [10] N. Mahammed, S. M. Benslimane, An evolutionary algorithm based approach for business process multi-criteria optimization, *Inter. J. of Organizational and Collective Intelligence (IJOICI)* 7 (2017) 34–53.
- [11] K. Georgoulakos, K. Vergidis, G. Tsakalidis, N. Samaras, Evolutionary multi-objective optimization of business process designs with pre-processing, in: *2017 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2017, pp. 897–904.
- [12] G. Tsakalidis, K. Vergidis, Towards a comprehensive business process optimization framework, in: *2017 IEEE 19th Conf. on Business Informatics (CBI)*, volume 1, IEEE, 2017, pp. 129–134.
- [13] Y.-W. Si, V.-I. Chan, M. Dumas, D. Zhang, A petri nets based generic genetic algorithm framework for resource optimization in business processes, *Simulation Modelling Practice and Theory* 86 (2018) 72–101.
- [14] M. Comuzzi, Ant-colony optimization for path recommendation in business process execution, *J. on Data Semantics* 8 (2019) 113–128.
- [15] J. Kurniasih, E. Utami, S. Raharjo, Query optimization: A metaheuristics approach using modified memetics algorithm (mma), in: *Journal of Physics: Conf. Series*, volume 1254, IOP Publishing, 2019, p. 012011.
- [16] C. Shajahan, E. E. Subramanyan, S. Kumanan, N. Firoz, Design process sequencing in product development process using design structure matrix and particle swarm optimisation techniques, in: *J. of Physics: Conf. Series*, volume 1355, IOP Publishing, 2019, p. 012027.
- [17] M. Rekik, K. Boukadi, H. Ben-Abdallah, An end-to-end system to enhance business process performance through new outsourcing to the cloud strategies, *Inter. J. of Business and Systems Research* 13 (2019) 1–20.
- [18] R. C. G.-L. Neira-Tovar, Un enfoque para la optimizacion de procesos de negocio: Metodologia prototipo focus for the business processes optimization: A methodology prototype, *Daena: Inter. Journal of Good Conscience* 14 (2019) 364–383.
- [19] S. Deshmukh, M. Agarwal, S. Gupta, N. Kumar, Moea for discovering pareto-optimal process models: an experimental comparison, *Inter. J. of Computational Science and Engineering* 21 (2020) 446–456.
- [20] H. Li, S. Weng, J. Tong, T. He, W. Chen, M. Sun, Y. Shen, Composition of resource-service chain based on evolutionary algorithm in distributed cloud manufacturing systems, *IEEE Access* 8 (2020) 19911–19920.
- [21] R. Tanese, Parallel genetic algorithm for a hypercube, in: *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology, Cambridge, MA*, Hillsdale, NJ: L. Erlbaum Associates, 1987., 1987.
- [22] C. B. Pettey, M. R. Leuze, J. J. Grefenstette, A parallel genetic algorithm, in: *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, 1987, pp. 155–161.
- [23] Y. Chen, Y. Zhong, Automatic path-oriented test data generation using a multi-population

genetic algorithm, in: 2008 Fourth Inter. Conf. on Natural Computation, volume 1, IEEE, 2008, pp. 566–570.

- [24] A. Chassiakos, G. Rempis, Evolutionary algorithm performance evaluation in project time-cost optimization, *J. of Soft Computing in Civil Engineering* 3 (2019) 16–29.
- [25] X.-q. Shi, W. Long, Y.-y. Li, Y.-l. Wei, D.-s. Deng, Different performances of different intelligent algorithms for solving fjsp: A perspective of structure, *Computational intelligence and neuroscience* 2018 (2018).