Communication Papers of the 18th Conference on Computer Science and Intelligence Systems pp. 21–28
DOI: 10.15439/2023F3491
ISSN 2300-5963 ACSIS, Vol. 37

Population-less Genetic Algorithm? Investigation of Non-dominated Tournament Genetic Algorithm (NTGA2) for multi-objective optimization

Michał Antkiewicz, Paweł B. Myszkowski
Wrocław University of Science and Technology
Faculty of Information and Communication Technology
ul.Ignacego Łukasiewicza 5, 50-371 Wrocław, Poland
email: {michal.antkiewicz, pawel.myszkowski}@pwr.edu.pl

***Abstract*—This paper investigates the Non-dominated Tournament Genetic Algorithm (NTGA2) to examine how selection methods and population interact in solving multi-objective optimization problems with constraints. As NTGA2 uses tournament and GAP selections that link the current population and population' archive, the experiments' results show that the population role is significantly reduced in some cases. The study considers two benchmark problems: Multi-Skill Resource Constrained Project Scheduling Problem (MS-RCPSP) and Travelling Thief Problem. Moreover, the paper's experimental study consists of new instances for multi-objective MS-RCPSP to show some interesting results that, in some cases, the proposed Genetic Algorithm does not need population in the evolution process.**

I. Introduction

THE POPULATION in Genetic Algorithms (GA) plays a very important role. Too small a population size could significantly reduce the exploration and get stuck in local optima (a premature convergence). However, too large a population also blocks the evolution progress, where selection cannot efficiently do its work. The population size in GA also plays a crucial role in multi-objective optimization (MOO) problems, where the final results consist of a set of solutions (non-dominated, Pareto Front Approximations, PFA). GA applied to MO to be efficient can store all explored non-dominated solutions in the *archive*. Moreover, some methods use the *archive* set to select individuals under selection pressure, like Non-dominated Tournament Genetic Algorithm (NTGA2)[7]. It allows the current population to work on temporal solutions, where the *archive* stores all non-dominated solutions, which makes a "permanent" memory. Such phenomena exist in over-constrained MOO problems, where genetic operators could make an offspring individual worse, and additional space (population) could help. In this paper, two MO NP-hard problems with constraints – Travelling Thief Problem (TTP)[3] and Multi-Skill Resource-Constrained Project Scheduling Problem (MS-RCPSP)[8] – are examined to investigate how NTGA2 explores the solution landscape and effectively uses *archive* and population in individual selection.

The rest of the paper is organized as follows. In Sec.II, a short related work is given. The investigated MS-RCPSP and TTP problems are briefly defined in Sec.III. An investigated NTGA2 is given in Sec.IV. Sec. V includes experimental results of the proposed NTGA2 MOO. Lastly, the paper is concluded in Sec.VI.

II. Related works

Effective cooperation between population, *archive*, and GAP selection works in GaMeDE2 [1] - an enhanced Multi-Modal Optimization technique (GaMeDE[6]), inspired by NTGA2, where GAP operator was introduced. While GaMeDE2 simplified the algorithm, empirical research confirmed the importance of alternating between broad exploration using the *archive* and local optimization with the population. This is achieved by triggering local optimization when the number of newly discovered optimal solutions exceeds a threshold or reverting to *archive* sampling when further optimization becomes unfeasible. Although this presents some form of adaptive operator switching, the main drawback is the requirement of fine-tuning the threshold value. It has been indicated as the area for future work to develop a fully adaptive solution that eliminates the need for manual parameter specification. For this purpose, it is necessary to answer the question of how to effectively switch between population- and *archive*-based selection, and on what basis to make this decision.

The authors of the survey [13] designed the adaptation taxonomy scheme for GA. They highlight three aspects to be considered: adaptation objects, adaptation evidence, and adaptation methods. Adaptation objects refer to the components within a GA. These objects include control parameters (crossover or mutation probability), evolutionary operators, and other elements. Adaptation evidence determines the basis on which adaptation occurs within a GA. There are four categories of adaptation evidence highlighted by the authors: deterministic factors, fitness values, population distribution, and combinations of fitness values and population distribution. There are several adaptation methods that might be implemented, including simple rules-based heuristics, or co-evolution. This paper examines how the structure and constraints of the problem affect the population- and *archive*-

Thematic track: Computational Optimization

based selection with a view to using it for the phase switching control.

III. Problem(s) definition

Two practical multi-objective problems have been selected to investigate the NTGA2 method. Both TTP and MS-RCPSP are constrained NP-hard problems with near-to-real-world combinatorial landscapes. Results have been compared using standard HyperVolume (HV, see Sec.V-B) measure.

A. Multi-Skill Resource-Constrained Project Scheduling Problem

The MS-RCPSP is a specific case of combinatorial NP-hard scheduling problems. It encompasses two interrelated sub-problems: task sequencing, and resource assignments. The objective of the MS-RCPSP is to determine a schedule that is both feasible and satisfies all the defined constraints. This involves assigning available resources to tasks and arranging the tasks on a timeline. In order for a schedule, denoted as PS, to be considered feasible, it must adhere to a predetermined set of constraints.

Each resource is connected to its salary r_{salary}, no salary can be negative. The set of skills S_r possessed by the resource r cannot be empty. Each task's duration d_t and finish time F_t are not negative. Tasks are constrained by a precedence relation – all task's predecessors must be finished before work on it can be started. All tasks must have assigned exactly one resource.

The skill extension of the MS-RCPSP is described in Eq. 1. The resource must have the skill at the required level or higher if it is assigned to a task.

$$\forall_{t\in T^r}\ \exists_{s_r\in S^r}\ h_{s_t} = h_{s_r} \wedge l_{s_t} \leq l_{s_r} \tag{1}$$

where T^r is a set of tasks assigned to a resource r, s_t is the skill required by the task t, S^r is the set of skills possessed by the resource r, h and l are the type and level of the skill respectively.

The latest definition of the MS-RCPSP is a many-objective optimization problem with five objectives (see [7][8]). The original two objectives – schedule duration (makespan) and cost – can be defined by Eq.2 and Eq.3. Further MS-RCPSP objectives tackle specific project scheduling aspects: average cash flow, skill overuse, and the average use of resources. The **Makespan** $\mathbf{f}_\tau(\mathbf{PS})$ of the project schedule PS is given as Eq.2.

$$f_\tau(PS) = \max_{t\in T} t_{finish} \tag{2}$$

where T is a set of all tasks, t_{finish} is the finish time of the task t. The **Cost** of the schedule is $\mathbf{f_C}(\mathbf{PS})$ defined as Eq. 3.

$$f_C(PS) = \sum_{i=1}^{n} R_i^{salary} * T_i^{duration} \tag{3}$$

where n is the number of all task-resource assignments, R_i^{salary} is the salary of a resource of the i'th assignment, $T_i^{duration}$ is the duration of the task of the i'th assignment.

The MS-RCPSP originally optimises 2-objectives $\mathbf{f}_\tau(\mathbf{PS})$ and $\mathbf{f_C}(\mathbf{PS})$, where all objectives must be minimized:

$$\min f(PS) = \min\left[f_\tau(PS), f_C(PS)\right] \tag{4}$$

B. Travelling Thief Problem

The TTP is a combination of two well-known optimization problems: the Traveling Salesman Problem (TSP) and the Knapsack Problem (KNP). A collection of cities is given, each characterized by its geographical coordinates, along with a set of associated items. These items are characterized by their weight and profit values. The objective is to determine an optimal route that visits all the cities while simultaneously selecting items from certain cities. The primary objective of the TTP, as expressed by Eq.5, encapsulates the main goal of this problem.

$$\min f(\pi, z) = \min f_\tau(\pi, z), \max f_P(z) \tag{5}$$

where π and z are the permutations of cities visited and the picking plan. The objective f_τ is to minimize the **traveling plan**. The f_P objective is the **profit** maximization based on the picked items. The relation between those problems is that picking items decreases travel speed.

$$f_\tau(\pi, z) = \sum_{i=1}^{n-1} \frac{d_{\pi_i,\pi_{i+1}}}{v(w(\pi_i))} + \frac{d_{\pi_n,\pi_1}}{v(w(\pi_n))} \tag{6}$$

where $d_{\pi_i,\pi_{i+1}}$ denotes the distance between two consecutive cities, n is a number of cities, $v(w(\pi_i))$ is the velocity in city π_i, which depends on weight w. As items are selected, the entire travel duration, as indicated by Eq. 6, undergoes modifications, and the velocity decreases in accordance with Eq. 7.

$$v(w) = v_{max} - \frac{W_c}{W}(v_{max} - v_{min}) \tag{7}$$

where W_c and W are the current and maximum allowed weights. The model defines the speed: maximum v_{max} and minimum v_{min} speed depending on W. The weight w is the accumulated sum of items picked up so far.

$$f_P(z) = \sum_{j=1}^{m} z_j z_j^{profit} \tag{8}$$

where m is the number of items, z_j is equal to 1 if the j'th item has been picked, 0 otherwise. z_j^{profit} is the profit of the j'th item.

The Eq.8 defines the profit as the second TTP objective. Furthermore, in order for the picking plan to be considered feasible, it is imperative that KNP constraint, as represented by Eq.9, is satisfied.

$$\sum_{j=1}^{m} z_j z_j^{weight} \leq W \tag{9}$$

where m denotes the total number of items, while z_j is defined as per Eq.8, and z_j^{weight} represents the weight of the j'th item. The aforementioned equation guarantees that the cumulative weight of the selected items remains within the W, which denotes the maximum permissible weight of the knapsack.

IV. Method

In this section, the NTGA2 method is introduced, and a greedy-based algorithm (see Sec.1) is used to get *feasible* solution in the MS-RCPSP problem.

A. Greedy–based Schedule Builder

Each investigated metaheuristic in this work to solve MS-RCPSP uses a greedy–based Schedule Builder to build the *feasible* schedule – see Algorithm 1 [7]. The method processes the tasks (in the given order). First, tasks with successors, then other tasks. The main goal is to assign each task at the earliest possible time it can be started. Namely, it is when all the predecessors of the tasks are finished, and its assigned resource finished its previous task assignment.

Algorithm 1 Greedy Schedule Builder for MS-RCPSP

for task t **do**
  $predEnd = maxFinish(t.predecessors)$
  $resEnd = t.getResource().getFinish()$
  $t.start = max(predEnd, resEnd)$
end for

B. Non-Dominated Tournament Genetic Algorithm 2

NTGA2[7] is an evolutionary metaheuristic promoting diversity by utilizing a *Gap* selection (GS) operator. GS works in the objective space, favoring the least explored parts of the *archive* – *Gap* in detail is given below. NTGA2 uses *archive* to store all non-dominated solutions and actively use it – see Algorithm 2. Firstly, NTGA2 initializes the population (usually a random one – see line 2). Then all individuals are evaluated (separately by each objective), and then $UpdateArchive$ takes place, where all non-dominated already found individuals are added and just dominated ones are removed.

The main loop starts (line 5) and repeats $Generations$ times. Each generation starts with a selection of individuals to the new population P_{next}. GS and the second selection (Pareto-dominance tournament selection) is used. The $gsGenerations$ parameter (line 8) switches selections a decides which one is used in the current generation. Line 15 presents the clone elimination mechanism used in NTGA2 to keep diversity in the population at a high level. Lastly, the genetic operators (e.g. mutation and crossover) should be specialized per problem. However, they can default to standard *single-point* crossover and *random bit* mutation.

The **Gap Selection** (GS) operator [7] aims to increase the *diversity* in *archive*. It operates in an objective space and considers each objective separately. The authors decided to select objectives as follows: offspring generation is divided into m

Algorithm 2 Pseudocode of NTGA2 [7]

1: $archive \leftarrow \emptyset$
2: $P_{current} \leftarrow GenerateInitialPopulation()$
3: $Evaluate(P_{current})$
4: $UpdateArchive(P_{current})$
5: **for** $i \leftarrow 0$ to $Generations$ **do**
6: $P_{next} \leftarrow \emptyset$
7: **while** $|P_{next}| < |P_{current}|$ **do**
8: **if** i mod (2 * $gsGen$) $< gsGen$ **then**
9: $Parents \leftarrow Tour\_selection(P_{current})$
10: **else**
11: $Parents \leftarrow Gap\_selection(Archive)$
12: **end if**
13: $Children \leftarrow Crossover(Parents)$
14: $Children \leftarrow Mutate(Children)$
15: **while** P_{next} contains $Children$ **do**
16: $Children \leftarrow Mutate(Children)$
17: **end while**
18: $Evaluate(Children)$
19: $P_{next} \leftarrow P_{next} \cup Children$
20: $UpdateArchive(Children)$
21: **end while**
22: $P_{current} \leftarrow P_{next}$
23: **end for**

parts (as the number of objectives), where each objective is selected during the corresponding part. It starts by calculating the "gap" size for each individual in the *archive*. It is calculated considering the two neighbor individuals using the minimal *Euclidean* distance. Those are the closest individuals, one with a worse objective value and one with a better value. The $GapValue$ is used as the Euclidean distance to the farther of those two neighbors. Additionally, individuals at the "edge" (i.e., the highest and lowest objectivities' values) of the *archive* have this distance set to an $infinity$ value, which is favored in selection.

Thus, the GS uses a tournament selection, considering $GapValues$ instead of $fitness$ directly. In this way, GS is more likely to select those individuals that lie close to the largest "gaps" in the *archive* and also promote the spread of the result *archive*. The second parent is selected as the $random$ neighbor of the first individual. For the individuals lying on the "edge" of PF approximation, it is possible that a second parent will not be selected. It will be selected similarly to the first one.

V. Experiments

The main goal of conducted experiments is to investigate further the effectiveness of the GS operator inside the NTGA2 method, applied to different scenarios. It can be hypothesized that its effectiveness varies depending on the problem, and further - instances. To carry out the structured experiments, the following **Research Questions** have been developed:

- ***RQ0.*** How the $gsGen$ (Gap Selection %) parameter affects the effectiveness of the NTGA2 method in bi-objective MS-RCPSP?
- ***RQ1.*** How do the characteristics (size, number of constraints) of the instance affect the effectiveness of the NTGA2 method in bi-objective MS-RCPSP?
- ***RQ2.*** Are the observations made for the TTP consistent with those for the MS-RCPSP?
- ***RQ3.*** Does the size of the computational budget affect the effectiveness of the Gas Selection in the NTGA2 method?
- ***RQ4.*** What aspects (differing or connecting two problems) can be used to adapt $gsGen$ parameter control?

A. Instances

In experiments, the iMOPSE dataset [7][8] is used. The original suite contains 36+6 MS-RCPSP instances created using real-world scheduling problems. All instances have varying tasks, resources, and skills to define problems. The final suite used in this paper contains 3 small and 6 randomly selected instances from the original set. Furthermore, several new instances were prepared using the iMOPSE generator to show the influence of constraints (e.g. introducing extreme low and high values for precedence relations or no skill requirements) and a number of tasks for NTGA2 effectiveness (e.g. 500 and 1000)[1].

For TTP, the benchmark dataset [2] has been selected - 16 instances differ in varying items per city (between 51 to 100). They could be divided into three groups that show the correlation between weights and profits of items: (1) with a strong correlation, (2) completely uncorrelated, and (3) with similar weights.

B. Quality measure of multi–objective optimisation

The most popular multi-objective metric is HyperVolume (HV) [9] – measures the diversity and convergence of the Pareto Front Approximation (PFA) that includes all non-dominated solutions calculated by a given method.

Results are normalized using the $NadirPoint$ - worst possible values for all objectives. For the MS-RCPSP: makespan – total sum of all tasks' duration; cost – the cost of schedule, where the most expensive resource performs all tasks. For the TTP: time – is twice the minimum time value; profit – equal to 0. On the other side - the $Ideal\ Point$ is the point with the best possible values for all objectives. For MS-RCPSP: makespan – duration of the shortest task multiplied by the number of tasks, divided by the number of resources; cost – the cost of the schedule, where all tasks are assigned to the cheapest resource. For TTP: time – the total length of the minimum spanning tree divided by the maximum speed; profit – achieved by a brute-force algorithm starting from the items with the highest profit/weight ratio.

[1]All used MS-RCPSP instances and gained results are published in http://imopse.ii.pwr.edu.pl

C. Reference methods

For a more comprehensive presentation of NTGA2 results, results of the state-of-the-art and best-known multi-objective optimization methods should also be considered.

Non-Dominated Sorting Genetic Algorithm II (**NSGA-II**) [5] is the classical method proposed in the year 2002 for MOO, utilizing the population sorting by $rank$ and $crowding\ distance$. The Strength Pareto Evolutionary Algorithm 2 (**SPEA2**) [11], a well-established method for MOO, employs environmental selection to enhance the exploration of the Pareto front. The Multi-objective Evolutionary Algorithm Based on Decomposition (**MOEA/D**) [12] is an evolutionary computation method designed for solving MOO problems by decomposing problems into a set of scalar sub-problems that are concurrently optimized.

D. Configurations

For all methods, the 5-Level Taguchi Parameter Design [10] was employed to fine-tune the parameters systematically. The best-found configurations used as the base values in the experiments are presented in Tab.I. Population Size ($PopSize$) is the constant number of individuals in a generation. For the MOEA/D, population size is derived from the number of decomposition vectors achieved using [4] algorithm for the given number of partitions ($PartNr$). The number of generations was adjusted to match the constant number of maximum births/fitness evaluations, for the MS-RCPSP computational budget was set to 50.000. For the TTP it was set to 250.000. Mutation probability (P_m) is a probability in $[0, 1]$. In the MS-RCPSP, it represents a chance of a single gene's random mutation. For the TTP, it is described using two different values: the chance of the random path segment being reversed; and the chance of a random item decision change ($bitflip$). Crossover probability (P_x) is the probability of two individuals crossover. All implemented methods use the same *Uniform crossover* operator for the MS-RCPSP and a combination of OX (route) with SX (knapsack) for the TTP. Tournament Size ($TourSize$) is the number is individuals considered whenever the tournament selection operator is used. Based on the original NTGA2 implementation, 2 values were found to be used, first for the Standard Tournament and second for the Gap tournament selection. The neighborhood Size ($NhSize$) is the number of adjacent decomposition vectors considered by MOEA/D when solutions are compared.

A number of generations ($gsGen$, see Tab.I) is the selection switch parameter used by the NTGA2. In the original NTGA2 paper, it has been interpreted as the number of generations that has to pass for the selection to switch, and it was set to 50. Although this parameter originally referred to the frequency of changes, a value of 50 can also be interpreted as 50 per 100 generations using the GS (Gap Selection %). Therefore, further in this paper, $gsGen$ is evaluated as the number of consecutive generations using the GS per 100 generations.

TABLE I
THE BEST FOUND CONFIGURATIONS FOR INVESTIGATED METHODS

MSRCPSP	PopSize	P_m	P_x	TourSize	gsGen	NhSize	PartNr
NTGA2	50	0.01	0.6	6 / 20	50*		
MOEA/D	(50)	0.015	0.2			6	50
SPEA2	200	0.015	0.99				
NSGA-II	300	0.015	0.99	2			

TTP	PopSize	P_m	P_x	TourSize	gsGen	NhSize	PartNr
NTGA2	50	0.9 / 0.9	0.3 / 0.3	6 / 20	50*		
MOEA/D	(100)	0.4 / 0.3	0.5 / 1.0			3	100
SPEA2	100	0.4 / 0.3	0.1 / 0.8				
NSGA-II	300	0.4 / 0.7	0.9 / 0.3	2			

E. Experimental procedure

The research environment with the NTGA2 method and additional reference methods have been implemented in Java based on the literature: MOEA/D, SPEA2, and NSGA-II. Additionally, some methods (e.g. MOEA/D) use reference points in calculations (objectives normalization). Others do not require them (like NTGA2) as they treat objectives separately. Reference points ($NadirPoint$ and $IdealPoint$) calculations as presented in the Sec. V-B.

The result of each run is a set of non-dominated solutions found for a given instance. Solutions are saved using absolute coordinates in the objective space. The experimental results have been evaluated on all selected instances for MS-RCPSP and TTP. Due to the non-deterministic nature of evolutionary computation, all runs have been repeated 30 times, and results have been averaged. To verify the statistical significance of the presented results Wilcoxon signed-rank test is used with $p\ value = 0.05$. A simple average is not an appropriate solution as HV strongly differs across the instances. Therefore, a ranking system has been applied to compare configurations and methods in all conducted experiments. The procedure starts with descending sort by the average HV and assigning the best rank (1) to the first configuration. Then, each configuration is considered subsequently. If its result is not significantly lower than the best of the current setup, it gets assigned the same rank. If the rank is significantly lower - the rank is incremented and assigned to this configuration. Each experiments table contains three summary rows at the bottom: average rank, median rank, and the dominance information ($+$ sole-best / $\sim$ co-best / $-$ worst).

F. Results for MS-RCPSP

To address the $RQ0$, five variants with different values of the $gsGen$ parameter were examined. The configurations utilized Gap selection in 0%, 25%, 50%, 75%, and 100% of generations, respectively. The results are presented in Tab.II.

As observed in Tab.II, the configuration employing 100% GAP selection clearly dominates in nearly all original instances. However, in the case of two small instances (15_6_10_6, 15_9_12_9), only GS from the *archive* does not yield the best results. Similarly, configuration alternating two selection methods prove to be the most effective for new instances with a high number of constraints (e.g., 100_10_4096_15). While the differences are statistically significant, they are very small.

The newly added instances (with suffix _0_0), devoid of skill constraints and task orders, did not introduce noticeable changes. It can be assumed that they are sufficiently similar to the existing cases. Considering the number of precedence relations, where the maximum theoretical number of direct relationships is $n * (n-1)/2$, for 100 tasks, it amounts to 4950. Therefore, the highest number of constraints in the set, which is 145, is still very small. Hence, the absence of constraints does not differ significantly. On the other hand, including instances with precedence relations at the level of several thousand introduce interesting cases that have not been observed before.

Fig. 1 indicates that, for dense PFA, there is no need to employ a population. It is easy to transition between solutions as they are close to one another. There is a large number of solutions that exploration based on the *archive* alone is sufficient, at least until a certain point. However, relying solely on the *archive* may become inadequate if the search space is highly constrained and all the 'low-hanging apples' are found. Theoretically, it is still unnecessary. In the current encoding, any feasible solution can transition to another in a single generation (as each gene can be modified independently), but it might not be very probable.

Using a population allows for delving "deeper" into certain areas of the sparse Pareto Front, as visible in Fig. 2. Classically, this brings about a solution to the problem of balance between exploration and exploitation. It is well illustrated in Fig. 3 containing results for the biggest instance in the suite. None of the configurations have sufficiently searched the space yet. The 'population-only' approach focused on a particular area, while the '*archive*-only' covered a wider range. As both approaches perform their role well, the latter achieves significantly better HV. It would likely be beneficial to activate the population search when relying on the *archive* ceases to yield progress instead of static parameters. To answer the second $RQ1$, the effectiveness does not explicitly depend on the instance size. However, rather constraints density and it changes over time as the *archive* saturates. This claim is supported by the results, where a lower budget (25.000, half of the original) results in better ranks for the '*archive*-only' approach.

Experimental results presented in this section showed that Gap affect the effectiveness of the NTGA2 applied to MS-RCPSP. How does such a mechanism work for TTP?

G. NTGA2 results for TTP

In order to verify if similar results can be observed for the TTP ($RQ2$), analogous experiments have been conducted using five configurations, which utilize GS in 0%, 25%, 50%, 75%, and 100% of generations, respectively. The results are presented in Tab.IV.

Compared to the MS-RCPSP, results achieved for the TTP (see Tab.IV) are more balanced, and there is no visible dominance of either configuration. The higher usage of GS has

TABLE II
RESULTS (HV) OF MS-RCPSP WITH 50K BUDGET

	Gap Selection Usage (%)				
instance	0	25	50	75	100
15_3_5_3	**1** (0.320398±1.67e-16)	**1** (0.320398±1.67e-16)	**1** (0.320398±1.67e-16)	**1** (0.320398±1.67e-16)	**1** (0.320398±1.67e-16)
15_6_10_6	**1** (0.545211±8.31e-06)	**1** (0.545213±2.51e-06)	**1** (0.545211±4.76e-06)	2 (0.545207±1.19e-05)	3 (0.544863±8.89e-05)
15_9_12_9	**1** (0.595148±1.30e-04)	**1** (0.595115±1.75e-04)	**1** (0.595092±2.05e-04)	2 (0.595009±2.58e-04)	3 (0.594575±2.96e-04)
100_5_20_9_D3	4 (0.429317±3.45e-03)	3 (0.435507±2.45e-03)	2 (0.436230±2.94e-03)	2 (0.436880±2.22e-03)	**1** (0.438390±1.94e-03)
100_5_48_9	3 (0.171241±2.63e-03)	2 (0.175926±8.50e-04)	**1** (0.176215±9.45e-04)	**1** (0.176347±8.66e-04)	**1** (0.176715±9.23e-04)
100_10_65_15	3 (0.458327±2.84e-03)	2 (0.469056±1.98e-03)	2 (0.469367±1.80e-03)	**1** (0.470235±1.74e-03)	**1** (0.470540±1.68e-03)
100_20_0_0	5 (0.683409±5.12e-03)	4 (0.726044±2.19e-03)	3 (0.733052±1.55e-03)	2 (0.734951±1.51e-03)	**1** (0.736693±1.19e-03)
100_40_0_0	5 (0.656597±7.34e-03)	4 (0.728598±5.49e-03)	3 (0.752332±3.74e-03)	2 (0.763022±2.99e-03)	**1** (0.769273±2.62e-03)
100_10_4096_9	2 (0.181034±2.51e-05)	**1** (0.181041±3.64e-06)	**1** (0.181042±2.93e-06)	2 (0.181039±5.45e-06)	3 (0.181026±7.46e-06)
100_10_4096_15	4 (0.250829±7.23e-05)	3 (0.250859±3.14e-05)	**1** (0.250865±0.00e+00)	**1** (0.250865±0.00e+00)	2 (0.250859±4.42e-06)
100_20_1024_9	4 (0.517370±2.50e-03)	3 (0.527683±9.73e-04)	**1** (0.528931±3.82e-04)	**1** (0.528962±3.47e-04)	2 (0.528699±2.60e-04)
100_20_2048_15	4 (0.380175±5.93e-04)	**1** (0.380732±2.20e-05)	**1** (0.380737±1.16e-05)	2 (0.380730±1.32e-05)	3 (0.380689±3.39e-05)
100_20_4096_9	3 (0.268188±9.21e-05)	**1** (0.268304±1.78e-06)	**1** (0.268299±3.12e-05)	**1** (0.268305±1.92e-06)	2 (0.268295±4.01e-06)
100_20_4096_15	3 (0.259517±1.60e-04)	2 (0.259638±9.18e-05)	**1** (0.259664±4.59e-05)	**1** (0.259678±3.28e-05)	2 (0.259650±3.17e-05)
100_40_1024_9	4 (0.570528±8.13e-04)	3 (0.574741±5.45e-04)	2 (0.575724±4.14e-04)	**1** (0.576005±3.15e-04)	**1** (0.576010±2.64e-04)
200_10_84_9	5 (0.636125±2.91e-03)	4 (0.670534±2.13e-03)	3 (0.678506±1.42e-03)	2 (0.681494±1.19e-03)	**1** (0.682893±1.26e-03)
200_20_97_9	5 (0.629058±7.46e-03)	4 (0.696804±4.35e-03)	3 (0.713753±2.85e-03)	2 (0.720315±1.71e-03)	**1** (0.724045±1.59e-03)
200_20_145_15	5 (0.571673±4.72e-03)	4 (0.617373±3.41e-03)	3 (0.627285±2.49e-03)	2 (0.630914±1.26e-03)	**1** (0.632386±1.49e-03)
200_20_0_0	5 (0.649978±5.06e-03)	4 (0.737694±5.42e-03)	3 (0.763551±4.12e-03)	2 (0.778068±2.52e-03)	**1** (0.784937±2.58e-03)
200_40_0_0	5 (0.719199±5.39e-03)	4 (0.778408±4.13e-03)	3 (0.798334±3.72e-03)	2 (0.808493±2.45e-03)	**1** (0.815440±2.66e-03)
500_10_512_5_A	5 (0.398676±1.58e-03)	4 (0.412768±1.24e-03)	3 (0.418172±1.13e-03)	2 (0.421138±7.19e-04)	**1** (0.422460±8.19e-04)
500_10_2048_5_A	5 (0.536240±1.56e-03)	4 (0.555732±1.62e-03)	3 (0.563177±1.46e-03)	2 (0.567528±1.55e-03)	**1** (0.570074±1.23e-03)
500_20_512_5_A	5 (0.553193±3.07e-03)	4 (0.587305±3.20e-03)	3 (0.601718±2.95e-03)	2 (0.609410±2.19e-03)	**1** (0.614964±1.90e-03)
500_20_0_0	5 (0.591966±4.23e-03)	4 (0.643410±3.08e-03)	3 (0.660743±3.52e-03)	2 (0.668717±2.71e-03)	**1** (0.675124±2.36e-03)
500_40_0_0	5 (0.619020±5.67e-03)	4 (0.681662±4.88e-03)	3 (0.705443±3.41e-03)	2 (0.717776±3.53e-03)	**1** (0.725970±3.46e-03)
1000_20_1024_5_A	5 (0.585542±3.56e-03)	4 (0.617928±3.22e-03)	3 (0.634121±3.98e-03)	2 (0.644301±4.14e-03)	**1** (0.650263±2.98e-03)
1000_20_4096_5_A	5 (0.564774±3.19e-03)	4 (0.583085±2.47e-03)	3 (0.594726±2.26e-03)	2 (0.599670±2.44e-03)	**1** (0.600960±2.24e-03)
1000_40_1024_10_A	5 (0.506093±4.80e-03)	4 (0.545322±4.19e-03)	3 (0.565749±4.58e-03)	2 (0.580091±4.65e-03)	**1** (0.587114±4.27e-03)
1000_20_0_0	5 (0.450345±3.67e-03)	4 (0.503184±4.13e-03)	3 (0.527720±3.98e-03)	2 (0.540095±3.83e-03)	**1** (0.549674±3.52e-03)
1000_40_0_0	5 (0.535800±5.89e-03)	4 (0.594065±4.69e-03)	3 (0.619015±4.71e-03)	2 (0.633995±4.26e-03)	**1** (0.643587±3.53e-03)
avg rank	4.067	3.067	2.233	1.733	1.4
med rank	5	4	3	2	1
dominance	(+0/∼ 3/−27)	(+0/∼ 6/−24)	(+0/∼ 10/−20)	(+0/∼ 8/−12)	(+18/∼ 4/−8)

TABLE III
METHODS COMPARISON (HV) OF MS-RCPSP WITH 50K BUDGET

method	ntga2 0%	ntga2 50% [7]	ntga2 100%	moea/d	nsgaii	spea2
avg rank	4.1	2.033	1.367	2.033	4	3.7
med rank	4	2	1	2	4	4
+	0	5	12	3	0	0
∼	3	6	8	7	1	1
−	27	19	10	20	29	29

a slightly better average ranking, but none reaches above 2.5 rank. This is most likely due to the difference in the encoding. Permutation-based (ordering) genotype encoding with inverse mutation does not allow for free transition between any solution – the transition from one solution to another might require multiple inverse operations on the genotype, which requires 'temporal' individuals - population.

Fig. 4 presents some similarities to the previous results. For the dense PFA, the best configuration uses an '*archive*-only' approach, which scans the space wider, while the 'population-only' is focused in a single direction. On the other end, Fig. 5 presents an instance with sparse PFA, where the 'population-based' approach significantly finds better results. Furthermore, to verify whether the budget has an impact on the GS effectiveness, other 'low-budget' experiments were carried out. Results achieved for lowered budget (50.000, i.e. one-fifth of the original). The effect is significant, as configuration for 75% GS improves from rank 2.5 to 2, and the 100% Gap configuration from 2.5 to 1.5. It supports the hypothesis of the increasing importance of 'population-based' selection (or decreasing importance of '*archive*-based' selection).

H. Summary

Experiments presented in previous sections showed that for MS-RCPCP and TTP, the computation budget plays an important role. For MS-RCPSP, a lower budget (25.000, half of the original) results in better ranks for the '*archive*-only' approach. Respectively, for TTP effect is also significant, as configurations that use the *archive* more 'frequently' (i.e. 75% or 100% Gap) improve their rank, which answers to $RQ3$.

Except for highly constrained instances, utilizing 100% GS yields the best results for the MS-RCPSP. This dominance of a single configuration is not that clear in the TTP. The potential reason could be the encoding difference since association encoding provides an easier transition between solutions than permutation encoding. Which is related to the constrainedness of the search space. The expected result for both problems is the better effectiveness of the 100% '*archive*-based' GS in less constrained instances - having dense PFA. In the most sparse PFA, 'population-based' selection provides a significant

TABLE IV
NTGA2 RESULTS (HV) OF TTP WITH 250K BUDGET

instance	Gap Selection Usage (%) 0	25	50	75	100
eil51_n50_bounded-strongly-corr_01	**1** (0.786307±2.22e-16)	5 (0.784464±0.00e+00)	4 (0.784500±2.22e-16)	2 (0.785861±1.11e-16)	3 (0.784563±2.22e-16)
eil51_n50_uncorr_01	5 (0.880994±1.11e-16)	4 (0.881789±2.22e-16)	**1** (0.884046±2.22e-16)	2 (0.883577±2.22e-16)	3 (0.881914±3.33e-16)
eil51_n50_uncorr-similar-weights_01	2 (0.732453±1.11e-16)	3 (0.731714±0.00e+00)	**1** (0.733751±3.33e-16)	4 (0.731639±2.22e-16)	5 (0.731452±2.22e-16)
eil51_n150_uncorr-similar-weights_01	4 (0.851645±2.22e-16)	**1** (0.856558±1.11e-16)	3 (0.855674±1.11e-16)	5 (0.851625±2.22e-16)	2 (0.856388±2.22e-16)
berlin52_n51_bounded-strongly-corr_01	**1** (0.884818±2.22e-16)	5 (0.880486±2.22e-16)	4 (0.881869±3.33e-16)	3 (0.883038±2.22e-16)	2 (0.883507±3.33e-16)
berlin52_n51_uncorr_01	2 (0.838511±1.11e-16)	5 (0.833224±2.22e-16)	3 (0.837703±2.22e-16)	4 (0.837138±1.11e-16)	**1** (0.838970±0.00e+00)
berlin52_n51_uncorr-similar-weights_01	**1** (0.722264±2.22e-16)	4 (0.720164±1.11e-16)	3 (0.720849±2.22e-16)	2 (0.721720±2.22e-16)	5 (0.719742±0.00e+00)
pr76_n75_bounded-strongly-corr_01	4 (0.813571±1.11e-16)	2 (0.818288±2.22e-16)	5 (0.813531±1.11e-16)	3 (0.816954±2.22e-16)	**1** (0.821745±0.00e+00)
pr76_n75_uncorr_01	5 (0.841923±2.22e-16)	3 (0.854713±1.11e-16)	**1** (0.858719±2.22e-16)	4 (0.854683±2.22e-16)	2 (0.855487±0.00e+00)
pr76_n75_uncorr-similar-weights_01	2 (0.767799±0.00e+00)	**1** (0.771500±1.11e-16)	5 (0.762926±0.00e+00)	3 (0.765856±1.11e-16)	4 (0.765030±1.11e-16)
kroA100_n99_bounded-strongly-corr_01	5 (0.866310±2.22e-16)	4 (0.874710±1.11e-16)	3 (0.881084±2.22e-16)	2 (0.884684±0.00e+00)	**1** (0.885007±3.33e-16)
kroA100_n99_uncorr_01	4 (0.844482±2.22e-16)	5 (0.838833±2.22e-16)	2 (0.852879±2.22e-16)	**1** (0.854227±3.33e-16)	3 (0.846971±0.00e+00)
kroA100_n99_uncorr-similar-weights_01	4 (0.886399±2.22e-16)	3 (0.887283±0.00e+00)	5 (0.883876±2.22e-16)	**1** (0.897105±0.00e+00)	2 (0.889732±2.22e-16)
rd100_n99_bounded-strongly-corr_01	5 (0.882900±2.22e-16)	**1** (0.892562±3.33e-16)	3 (0.889671±0.00e+00)	2 (0.892136±2.22e-16)	4 (0.889643±1.11e-16)
rd100_n99_uncorr_01	5 (0.851845±2.22e-16)	4 (0.856888±2.22e-16)	2 (0.857302±2.22e-16)	3 (0.857012±2.22e-16)	**1** (0.862809±2.22e-16)
rd100_n99_uncorr-similar-weights_01	4 (0.892621±3.33e-16)	5 (0.890683±2.22e-16)	2 (0.898613±2.22e-16)	**1** (0.898724±2.22e-16)	3 (0.893712±2.22e-16)
avg rank	3.375	3.438	2.938	2.625	2.625
med rank	4	4	3	2.5	2.5
dominance	(+3/∼ 0/−13)	(+3/∼ 0/−13)	(+3/∼ 0/−13)	(+3/∼ 0/−13)	(+4/∼ 0/−12)

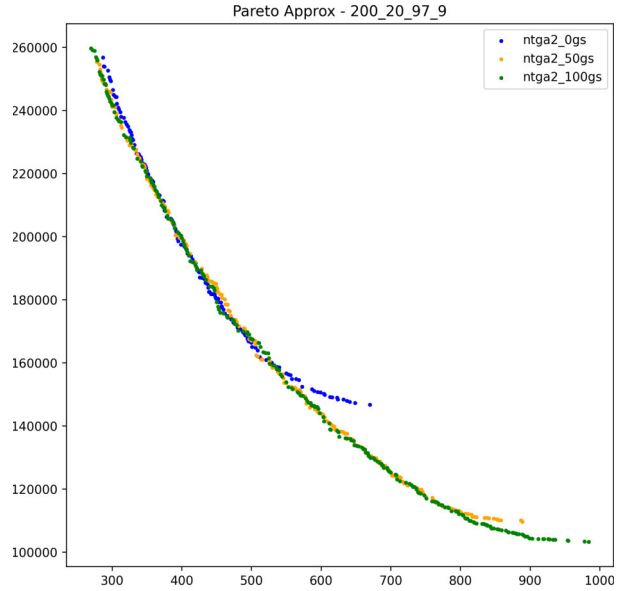

Fig. 1. Comparison of PFA for MS-RCPSP – 200_20_97_9 for GS configs.: 0%, 50% and 100%.

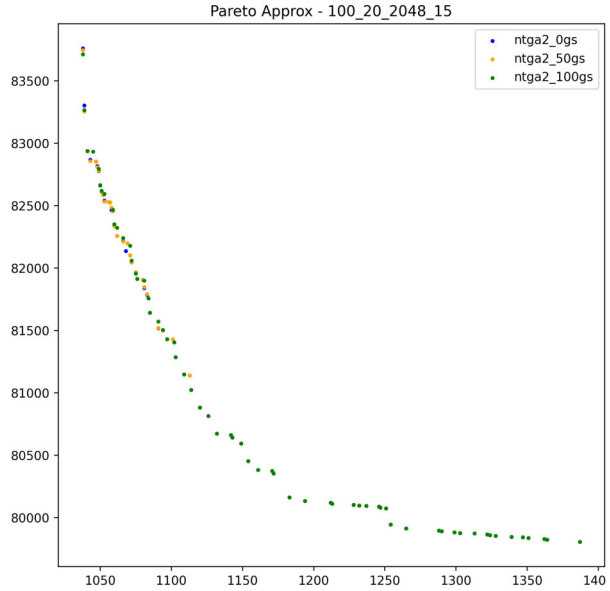

Fig. 2. Comparison of PFA for MS-RCPSP – 100_20_2048_15 for GS configs.: 0%, 50% and 100%.

TABLE V
METHODS COMPARISON (HV) OF TTP WITH 250K BUDGET

method	ntga2 0%	ntga2 50% [7]	ntga2 100%	moea/d	nsgaii	spea2
avg rank	1.938	1.312	1.312	3	3.75	2.812
med rank	2	1	1	3	4	3
+	0	2	5	0	0	0
∼	5	9	6	0	0	2
−	11	5	5	16	16	14

boost. Another potential cause is the computational budget vs the instance size. The importance of the 'population-based' approach improves over the execution time as the *archive* becomes more saturated. To answer the $RQ4$, the potential evidence for the $gsGen$ value adaptations are constraint density, encoding (transition freedom), and *archive* saturation. Where some or all of the above might be entangled.

VI. CONCLUSIONS AND FUTURE WORK

This paper shows the results of an investigation of how NTGA2 with Gap selection effectively uses *archive*s in solving multi-objective problems with constraints (MS-RCPSP and TTP). There are five answered research questions: the size of the problem instance and number and how aspects, and problems (TTP and MS-RCPSP) differ to determine the Gap (%) selection parameter. The main conclusion is that in some cases (instances), the *archive* (and the Gap selection)

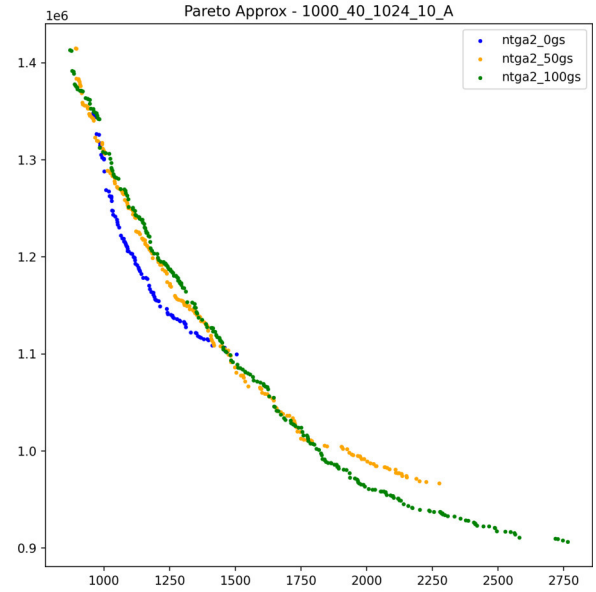

Fig. 3. Comparison of PFA for MS-RCPSP – 1000_40_1024_10_A

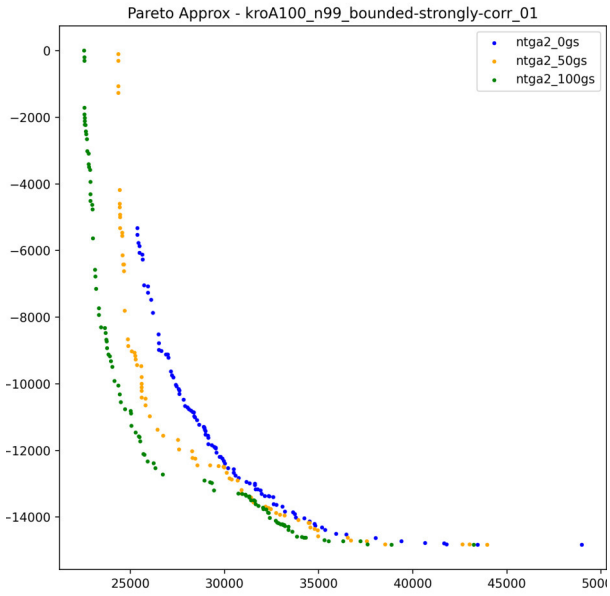

Fig. 4. Comparison of PFA for TTP – kroA100_n99_bounded-str.-corr_01

plays a crucial role, and the population could be eliminated. Experimental results presented a correlation between the Gap selection effectiveness and constraints density, as well as optimization progress.

There are several promising future directions of research. The GS in most cases (TTP and MS-RCPSP) prefers gap selection 100%, but in some cases (i.e. a large number of constraints) reduces to 50%. It encourages further work on Gap adaptation in NTGA2 to increase final NTGA2 effectiveness. Moreover, we empirically showed that such a situation occurs in two benchmark problems (TTP and MS-RCPSP), and plan

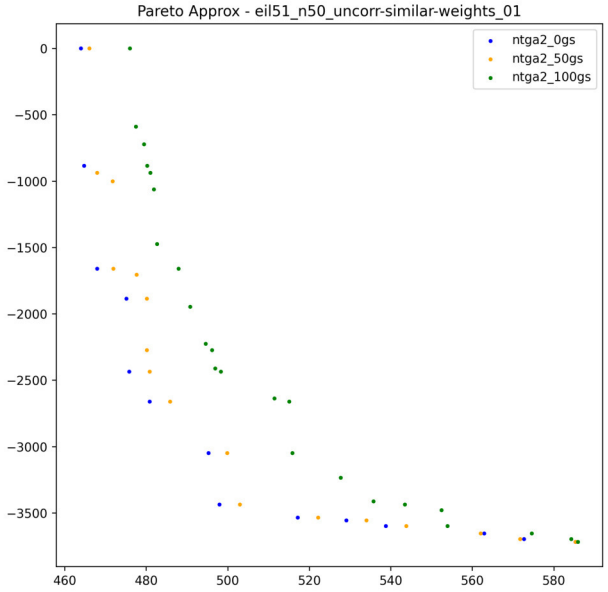

Fig. 5. Comparison of PFA for TTP – eil51_n50_uncorr-similar-weights_01

to investigate other multi-objective problems (including many-objective).

REFERENCES

[1] Antkiewicz, M., and Myszkowski, P.B, and Laszczyk M. "GaMeDE2—improved Gap–based Memetic Differential Evolution applied to Multimodal Optimization." 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS). IEEE, 2022.

[2] Blank, J., Deb, K. & Mostaghim, S. Solving the Bi-objective Traveling Thief Problem with Multi-objective Evolutionary Algorithms. *Evolutionary Multi-Criterion Optimization*. pp. 46-60 (2017)

[3] Bonyadi, M., Michalewicz, Z. & Barone, L. The travelling thief problem: The first step in the transition from theoretical problems to realistic problems. *2013 IEEE Congress On Evol. Comp.*. pp. 1037-1044 (2013)

[4] Das, I. & Dennis, J. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal On Optimization*. **8**, 631-657 (1998)

[5] Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions On Evolutionary Computation*. **6**, 182-197 (2002)

[6] M.Laszczyk, P.B. Myszkowski, "A gap–based memetic differential evolution (gamede) applied to multi–modal optimisation–using multi–objective optimization concepts", in: Intel. Inf. and Database Systems: 13th Asian Conf., ACIIDS 2021, Phuket (Thailand) 2021, Proc. 13, pp. 211–223.

[7] Myszkowski, P. & Laszczyk, M. "Diversity based selection for many-objective evolutionary optimisation problems with constraints". *Information Sciences*. **546** pp. 665-700 (2021)

[8] Myszkowski, P.B. & Laszczyk, M. "Investigation of benchmark dataset for many-objective Multi-Skill Resource Constrained Project Scheduling Problem". *Applied Soft Computing*. **127** pp. 109253 (2022)

[9] Myszkowski, P.B. & Laszczyk, M. Survey of quality measures for multi-objective optimization: Construction of complementary set of multi-objective quality measures. *Swarm And Evolutionary Computation*. **48** pp. 109-133 (2019)

[10] Vijayan, N. & Al. Taguchi's Parameter Design: A Panel Discussion. *Technometrics*. **34**, 127-161 (1992)

[11] Zitzler, Eckart, Marco Laumanns, and Lothar Thiele. "SPEA2: Improving the strength Pareto evolutionary algorithm." TIK-report 103 (2001).

[12] Zhang, Qingfu, and Hui Li. "MOEA/D: A multiobjective evolutionary algorithm based on decomposition." IEEE Transactions on evolutionary computation 11.6 (2007): 712-731.

[13] Zhang, Jun, et al. "A survey on algorithm adaptation in evolutionary computation." Frontiers of Electrical and Electr. Eng. 7 (2012): 16-31.