Diversity-Based Adaptive Genetic Algorithm for a Workforce Scheduling and Routing Problem

Haneen Algethami, Dario Landa-Silva
School of Computer Science, ASAP Research Group
The University of Nottingham, United Kingdom
Email: {haneen.algethami,dario.landasilva}@nottingham.ac.uk

Abstract—The Workforce Scheduling and Routing Problem refers to the assignment of personnel to visits across various geographical locations. Solving this problem demands tackling numerous scheduling and routing constraints while aiming to minimise total operational cost. One of the main obstacles in designing a genetic algorithm for this highly-constrained combinatorial optimisation problem is the amount of empirical tests required for parameter tuning. This paper presents a genetic algorithm that uses a diversity-based adaptive parameter control method. Experimental results show the effectiveness of this parameter control method to enhance the performance of the genetic algorithm. This study makes a contribution to research on adaptive evolutionary algorithms applied to real-world problems.

Keywords—*Genetic Algorithms, Adaptive Evolutionary Algorithm, Workforce Scheduling and Routing*

I. Introduction

The Workforce Scheduling and Routing Problem (WSRP) is described as the assignment of personnel to visits, requested by customers, across different geographical locations. This problem combines scheduling and routing problems, both of which are known to be NP-Hard [1]. The *scheduling aspect* of the problem assigns personnel to visits in order to fulfil work demands and other requirements. The *routing aspect* of the problem consists of generating routes for the workers to service customers across various locations within given time-windows. The objective is to minimise operational costs while attending the additional requirements expressed by customers, workers and the business. A type of WSRP arises in home health care where nurses and care workers should be assigned to visit patients in their homes in order to carry out some tasks, e.g. administering medication, monitoring serious illness and unstable health status and injections. Genetic Algorithms (GAs) have been shown to be effective approaches to find solutions for problems combining scheduling and routing where exact methods are less effective, e.g. [2]–[4]. A base-line GA was proposed by [5] that identified the best set of operators and parameters for each instance of WSRP. Despite its success in obtaining good solutions, the base-line GA performance was limited by the tuning of parameters being computationally expensive. Although parameter values can be tuned before the algorithm execution, different values might provide better results at different stages of the evolutionary process [6].

An adaptive parameter control approach that seeks to maintain diversity in the population of solutions is proposed here to enhance the performance of a genetic algorithm. The aim is to be able to tackle a wider variety of WSRP scenarios. A total of 42 instances from six different real-world home

health care (HHC) scenarios are used in the experiments here. The specific objectives of this paper are:

- To identify the features that contribute to the early convergence of the genetic algorithm.

- To propose six different variations of the adaptive parameter control method, with respect to the identified features, that aim to enhance algorithm efficiency.

- To analyses the impact of adaptive parameters on the feasibility of solutions and the algorithm speed.

In what follows, Section II reviews related work. Section III describes the WSRP, its formulation and the instances used in this paper. Section IV outlines the proposed algorithm. Section V presents the experimental study and discusses the obtained results. The paper is then concluded in Section VI.

II. Related Work

Recent research on the WSRP considered here is reviewed next. A mixed integer programming (MIP) with decomposition method [7] required considerable computation time (up to several hours) to solve larger problem instances with hundreds of tasks, indicating the need for faster solution methods. A Variable Neighbourhood Search (VNS) algorithm using problem-specific neighbourhood heuristics was presented in [8]. The VNS obtained high-quality solutions and in less computation time for the same set of problem instances used in [7]. An investigation was presented in [4], [9] comparing various genetic operators within two simple GAs to tackle the subject problem. A more efficient genetic algorithm with tuned parameters and using a customised solution representation to maintain feasibility of solutions was proposed in [5].

A number of studies have applied GAs to real-world problems where scheduling and routing are combined. Examples include [2], [3]. In those works, the focus has been on algorithm design to obtain good solutions. However, less attention has been given to developing an adaptive algorithm for this type of problems. There has been a number of works describing adaptive parameter control within evolutionary algorithms (EAs) in real-world applications [6], [10]. The study proposed by [11] used a method that incorporated global and local selections to maintain a balance of healthy individuals in the population. The effects of changing population size was investigated in [12]. Other works have used an adaptive process for managing dynamic populations in the algorithm. The study by [13] proposed to adaptively resize the GA population. This

paper investigates the process of dynamic population resizing tailored especially for WSRP.

The intended contribution of this work is to present an efficient diversity-based adaptive GA for tackling real-world WSRP instances. The proposed method is capable of producing competitive feasible solutions for a set of real-world instances of highly-constrained WSRP scenarios. The proposed GA harnesses the power of the adaptive parameter control method with problem-specific genetic operators to solve the WSRP instances considered.

III. PROBLEM DESCRIPTION

A WSRP solution is a daily plan of visits, i.e. a set of workers $W = \{w_1, w_2, \ldots, w_{|W|}\}$ assigned to perform a set of tasks $T = \{t_1, t_2, \ldots, t_{|T|}\}$ for customers at different locations. The assignment of a worker to travel to a customer location in order to perform a task is called a visit. Thus, $x_{i,j}^w$ is a binary decision variable that indicates if a path connects two nodes (visit i and visit j) or not. The assignment $x_{i,j}^w = 1$ means that worker w travels from visit i to visit j, thus w makes both visits. For visit j, if $x_{i,j}^w = 1$ then $y_j \leq r_j - 1$ where r_j is the *number of workers* required for visit j and y_j is an integer decision variable indicating the number of unsatisfied assignments, hence $\sum_{w \in W} \sum_{i \in T} \sum_{j \in T} x_{i,j}^w + y_j = r_j$.

This paper tackles a home health care (HHC) planning problem, in which workers are nurses, doctors, health carers, etc., and customers are patients receiving health care at their home. Several features have been identified as important in solutions to HHC scenarios, such as distance travelled and customers' and workers' requirements and preferences [14]. A good quality solution for an HHC planning problem should have low operational cost as well as all tasks being assigned in such a way as to satisfy the existing requirements. Table I lists the objectives and constraints in the WSRP considered here. See [7] for details of the MIP model of this WSRP. Note that in [7], unassigned visits is considered a soft constraint. However, here this is a hard constraint, hence all visits must be assigned. Additionally, time-conflict constraint is considered in this paper, while the study by [7] avoided time-conflicts. A time-conflict occurs when a worker is assigned to visits overlapping in time.

Table I.   OBJECTIVES AND CONSTRAINTS IN THE WSRP.

Objectives	Hard Constraints	Soft Constraints
Travelling Cost	Skills Requirements	Unassigned Visits
Payment Cost	Time-Conflicts	Area Availability
	Maximum Hours	Time Availability
		Preferences
		(skills,worker,area)

A solution S is a set of assignments of workers to make visits. The objective function includes the *operational cost* and the *penalty cost*. The operational cost includes wages plus journey costs for all workers and is given by the accumulated cost $d_{i,j} + p_j^w$, where $d_{i,j}$ is the distance travelled between visit i to visit j and p_j^w is the cost of assigning worker w to visit j. These costs are set by the service provider in the HHC scenarios used here. The penalty cost is the accumulated penalty for the violations of soft-constraints. An assignment can be written as a tuple $(x_{i,j}^w, y_j, a_j^w, \psi_j^w, \theta_j^w, \tau_j^w)$, where a_j^w is the arrival time of a worker w to the location of visit j. The assignment is also composed of binary decision

variables indicating an assignment of worker w to visit j with violations on area availability (ψ_j^w), time availability (θ_j^w) and conflicting assignments (τ_j^w). The non-satisfaction of preferences is also included in the penalty cost. There are three types of preferences including preferred worker-customer pairing, worker's preferred region and customer's preferred skills. There is a degree of satisfaction for these preferences when assigning a worker w to a task j and is given by ρ_j^w which has a value that ranges between $[0, 3]$. For each assignment, the satisfaction value for each preference ranges between $[0, 1]$, from not satisfied to satisfied. The satisfaction level is reverted to a penalty by subtracting it from the full satisfaction score, which is $3r_j$ for a visit j.

The best solution should have: the least *operational cost* and the least *penalty cost*. A weighted sum is proposed to combine the objectives into a single scalar value [5], [8]. The objective function is written as in equation (1), where weights $\lambda_1, \ldots, \lambda_5$ are defined to establish priority between objectives (more about the weights used here later in the paper).

$$
\begin{aligned}
f(S) = \lambda_1 &\sum_{w \in W} \sum_{i \in T} \sum_{j \in T} (d_{i,j} + p_j^w) x_{i,j}^w \\
+ \lambda_2 &\sum_{w \in W} \sum_{i \in T} \sum_{j \in T} (3r_j - \rho_j^w) x_{i,j}^w \\
+ \lambda_3 &\sum_{w \in W} \sum_{j \in T} (\psi_j^w + \theta_j^w) \\
+ \lambda_4 &\sum_{j \in T} y_j + \lambda_5 \sum_{w \in W} \sum_{j \in T} \tau_j^w \quad (1)
\end{aligned}
$$

IV. DIVERSITY-BASED ADAPTIVE GA (DAGA)

This section describes the proposed adaptive aspects of the GA as an extension to the approach proposed in [5]. The proposal here is an adaptive control mechanism for parameters values, particularly genetic operator rates P_c and P_m in addition to the population size M. The work presented in [5] showed that each WSRP instance has features that affected the selection of parameter values in the GA. The parameter values and genetic operators performance were dependent on the problem size reflected in the chromosome length. It was observed that an increase in the chromosome length required a smaller population size in the algorithm and vice versa. There was also an inverse relationship between the population size and the number of generations required for the GA to perform well. Therefore, in order to enhance the GA efficiency, an adaptive parameter control mechanism is proposed here. Adaptiveness of parameter values and genetic operator rates has been approached in the literature through rules [15] and randomness [16]. Hence, two elements are investigated in the adaptive parameter control proposed here. One element is feedback through adaptive rates and population resizing. The other element is randomly changing the crossover when no further improvements are observed. This method is used to delay the premature convergence by having diverse performances provided by different crossovers [17].

The proposed DAGA method works as follows. First, an initial population P of M individuals (one-day plans) is created based on the indirect chromosome encoding that

constructs feasible solutions. The representation is designed to include all assigned visits $|V|$, thus chromosome length vary according to the problem-size. For example, if instances are relatively small, a large M is required as a result of the large degree of similarity between individuals in P. For a large instance on the other hand there is less chance to have similar solutions, thus, a small M is sufficient. The population diversity has an effect on the setting of parameter values. Hence, a diversity-based scale is used to calculate the required change in the adaptive features. At the start of each generation, $9M/20$ pairs of parent individuals are selected using binary tournaments. Then, a new population is created through crossover and mutation. The suggested crossover is applied with probability P_c to each pair of parents to generate two offspring. Each offspring goes through a flat-cost flip (FCF) mutation, with some probability P_m. The operator performance influences the information feedback process where the adjustment of a parameter value is done. At the end of each generation t, the population is updated using an elitism strategy. The $M/10$ best individuals from the current population are kept and along with the $9M/20$ offspring individuals generated, the new population of M solutions is formed. After the new population is created, if there are no cost value improvements on the best so far solutions for a number of generations, the GA is considered to have stagnated. Thus, the updates to parameter values occur in response to the feedback process that seeks to maintain a diverse population to improve the effectiveness of the search. Such feedback from the search is used to generate the next population. Details of the diversity-based adaptive control approach to update parameter values during the GA's execution are described next.

A. Diversity Measurements

To measure the population diversity, a metric to measure the level of variation in a population was adopted from [18]. This method has also been useful for similar representation schemes used in the office space allocation problem [19]. The percentage of non-similarity within a set of chromosomes is given by equation (2), where $d_(j)$ is the number of different genes in the jth position and $|V|$ is the chromosome size for an individual in P. The diversity value for a population is then $D(P) = Variation_{(M)}/100$.

$$Variation_{(M)} = \frac{\sum\limits_{j=1}^{V} \frac{d_(j)-1}{(M-1)}}{V} \times 100 \qquad (2)$$

B. Search Operators Rates Updates

Initial parameter values were selected after an offline tuning method [5]. If the GA stagnates for a number of generations, crossover and mutation rates are adjusted according to the calculations presented by [11]. That adjusting approach is used due to its capability of adapting quickly on highly multi-modal fitness landscapes. A resetting process is also used for the adaptive rates P_c and P_m which forgets the rates history if the rates values are out of range. This forgetting step means ignoring all the previous feedback process that led up to the inflation of the adapted values. Details for the calculation of parameter rates are described next.

1) *Adaptive Crossover Rate P_c:* In general, crossover encourages the population to converge by reducing diversity. This is because after a number of generations, similarity between individuals increases as a result of only recombining existing genetic material. When similar individuals are recombined, *inbreeding* occurs and no additional diversity is added through crossover. Thus, a diversity-based adaptive parameter control is used to maintain a diverse population and avoid early convergence. Upper and lower values are identified for the control mechanism to generate values in that range. The adaptive crossover rate here ranges between $Q_1 \leq P_c \leq Q_2$, with $Q_1 = 45\%$ and $Q_2 = 100\%$.

$$P_c = \left[\frac{Diversity}{Diversity_{max}} * (Q_2 - Q_1) \right] + Q_1 \qquad (3)$$

The crossover rate P_c is then updated according to equation (3). The P_c value is updated with a value within the range, based on the population diversity, where $0 < Diversity \leq Diversity_{max}$. The population diversity influence is the ratio between the current diversity $Diversity$ and $Diversity_{max}$ provided this value does not exceed $Diversity_{max}$, which is the value of the population variance, i.e. how far each individual in the population is from the mean.

2) *Adaptive Mutation Rate P_m:* To introduce diversification into the population, mutation is usually applied. The selected range for the mutation rate P_m is $Q_1 \leq P_m \leq Q_2$, where $Q_1 = 10\%$ and $Q_2 = 60\%$. The overall mutation rate P_m is calculated with respect to the fitness improvements given by a mutation and the current diversity of the population. Thus, the new P_m is the average value of the calculations in equations (5) and (6) as stated in equation (4).

$$P_m = \frac{P_m^{Fitness} + P_m^{Diversity}}{2} \qquad (4)$$

Equation (5) calculates the mutation rate with respect to the diversity. A diverse population avoids early convergence, hence more individuals are explored. However, a highly converged population is subject to higher mutation rates. The value of P_m is equal to Q_1 when the population has the highest diversity given by $Diversity_{max}$, i.e. the population is almost in complete disruption.

$$P_m^{Diversity} = \frac{Diversity_{max} - Diversity}{Diversity_{max}} \times Q_1 \qquad (5)$$

To encourage further exploration, mutation is applied on individuals with low fitness. Equation (6) generates a value that uses the parent fitness $f(S)$, where $f(S)_{max}$ and $f(S)_{min}$ correspond to the best and worst fitness values of the individuals in the population respectively. The population is maximally diverged when the selected parents are the worst individual in the population.

$$P_m^{Fitness} = \frac{f(S)_{max} - f(S)}{f(S)_{max} - f(S)_{min}} \times Q_1 \qquad (6)$$

C. Dynamic Population Resizing

Preserving the population diversity is a key element for increasing the search coverage by performing an in-depth exploration of the fitness landscape [20]. The population diversity is affected by the population size. If the population size is too small the algorithm gets trapped in a sub-optimal region of the search space [21]. This is because the algorithm is unable to find better quality solutions as a result of inbreeding. Certain population size might be good at some point of the search but requires a change at different stages of the evolutionary process. Therefore, an adaptive population resizing is implemented here which works as follows. For population P, its size M is changed dynamically during the run when no improvement is observed. The deterministic change occurs with respect to the step size δ, only if M ranges between $100 \leq M \leq 550$. To preserve the best found solutions, the best $M/10$ individuals of the population are always kept. The value δ is the step-size which is relative to the current diversity $D_{Current}$ and the population size, that is, $\delta = D_{(Current)} \times M$.

Two methods of population resizing are used as stated in Equation (7). The first is to increase M by δ individuals in the next population. The addition occurs by constructing new individuals. The second case is to shrink M by removing the δ worst individuals from P.

$$M = \begin{cases} M + \delta & \text{if } D_{Current-1} < D_{Current-4}, \\ M - \delta & \text{if } D_{Current-1} > D_{Current-4}, \\ M & \text{if } D_{Current-1} = D_{Current-4}, \end{cases} \quad (7)$$

This method maintains the best individuals while maintaining the diversity level, to obtain better results and/or faster convergence. For example, when M is small, the diversity is lost between iterations. As a result, adding more individuals, regardless of their fitness value, promotes diversity and encourages exploration in the search. On the other hand, when M is large, the diversity is increasing throughout the generations. Thus, decreasing M allows the crossover to focus on better regions of the fitness landscape and henceforth computing results faster. All these changes are triggered by an increase or a decrease in the population diversity. If there is no change in the diversity, M remains the same. Equations (8) ensures keeping the population size within range.

$$M = \begin{cases} M + \delta & \text{if } M < 100, \\ M - \delta & \text{if } M > 550, \end{cases} \quad (8)$$

D. Random Crossover Selection

This method allows exploration of the solution space by using different operators when a particular operator fails to improve the search. The idea is to choose an arbitrarily crossover, if there has been no improvements for the last four generations. Different operators require different rates at each stage of the search. Therefore, further improvements are made on the global performance of the whole population by the adaptive crossover rates. The crossovers used in this study are: flat-cost (FCX) and partially-matched flat-cost crossovers

---

Algorithm 1 Diversity-Based Adaptive Control for the GA

Require: P_c, P_m, CrossoverOperator and M = ⟨ User Predefined ⟩
1: *Pick-random* := CrossoverOperator
2: **if** $P_c > 0.45$ AND $P_c < 1.0$ **then**
3:     Update P_c according to equation (3)
4: **else**
5:     Reset P_c
6: **end if**
7: **if** $P_m > 0.1$ AND $P_m < 0.60$ **then**
8:     Update P_m according to equation (4)
9: **else**
10:     Reset P_m
11: **end if**
12: **if** $M > 100$ AND $M < 550$ **then**
13:     Update M according to equation (7)
14: **else**
15:     Update M according to equation (8)
16: **end if**

(PMFCX) [5], single point (1PX), two point (2PX), uniform (UX) and Half Uniform (HUX) crossovers [22].

Algorithm 1 presents the diversity-based adaptive control for the GA. After the new population is created, if there is no improvement on the best so far solution for a number of generations, the GA is considered to have stagnated. Thus, the updates to the parameter values occur in response to the feedback process (according to the current population diversity) to maintain a diverse population when generating the next population. Initially, the crossover operator is changed by selecting a random crossover (step 1). Then, the crossover rate P_c and a mutation rate P_m are adjusted (steps 2-11) followed by the population resizing (step 12-16).

V. EXPERIMENTAL STUDY AND RESULTS

This section describes and compares several variations of the adaptive aspects against the previously established baseline GA presented in [5]. The algorithm variations investigated are: adaptive rates only (AGA), adaptive rates with dynamic population re-sizing (AGADP), dynamic population re-sizing only (GADP), adaptive rates with random crossover (RCAGA), random crossover with dynamic population re-sizing (RCGADP), adaptive rates and dynamic population size with random crossover (RCAGADP).

The parameter values are set for each groups of problem instances *before the run* as suggested by [5]. For **A, B**: P_m = 50%, P_c= 50%, M= 500. For **C**: P_m = 30%, P_c= 50%, M= 500. For **D, E**: P_m = 10%, P_c= 0.1, M= 100. In AGA, AGADP and GADP, a set of crossover operators were chosen for each problem set as follows. For **A, B** cost-based greedy crossover (CGX) is used. For **C** partially-matched greedy crossover (PMGreedyX) is used. For **D, E, F** flat-costs crossover (FCX) is used. During the run, parameters values were updated with respect to the population diversity as explained in Section IV. Based on preliminary tests, the parameter ranges are empirically set as follows: mutation rate $P_m \in [0.10, 0.55]$, crossover rate $P_c \in [0.45, 0.100]$ and the dynamic population size $M \in [100, 550]$. A random crossover is selected and applied over time when the algorithm is stuck at a local optima. To assess the statistical significance of the obtained results, each run was executed 8 times, seeded with the same initial population and all methods had the same

Table II.   MAIN FEATURES OF THE 42 HOME HEALTH CARE PROBLEM INSTANCES.

			A								B				
	01	**02**	**03**	**04**	**05**	**06**	**07**	**mean**	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**mean**
Number of Visits	31	31	38	28	13	28	13	26	36	12	69	30	61	57	61	46
Number of Workers	23	22	22	19	19	21	21	21	25	25	34	34	32	32	32	30
Number of Areas	6	4	5	4	4	8	4	5	6	5	7	5	8	8	7	7

			C								D				
	01	**02**	**03**	**04**	**05**	**06**	**07**	**mean**	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**mean**
Number of Visits	177	7	150	32	29	158	6	80	483	454	585	520	538	610	611	543
Number of Workers	1037	618	1077	979	821	816	349	813	164	166	174	174	173	174	173	171
Number of Areas	8	4	7	8	6	11	6	7	13	12	15	15	15	15	15	14

			E								F				
	01	**02**	**03**	**04**	**05**	**06**	**07**	**mean**	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**mean**
Number of Visits	418	425	462	351	461	301	498	416	1211	1243	1479	1448	1599	1582	1726	1470
Number of Workers	243	244	267	266	278	278	302	268	805	769	898	789	889	783	1011	901
Number of Areas	13	14	15	13	15	13	16	14	45	46	54	47	59	44	64	51

amount of computation time. The implementation was in Java running on a PC with I7 four-core processor with hyper-threading enabled and 16GB of RAM.

A. Problem Instances

Problem instances from three UK real-world HHC scenarios are used as instances of WSRP [1] in this study. There are 7 problem instances in each scenario for a total of 21 instances. Table II shows the main features of each problem instance. Scenario A instances are considered the smallest, while instances in scenario F are the largest. Problem instances in scenario C are very different to the instances in the other 6 scenarios by having a much larger number of workers than the number of visits.

B. Effect of Adaptive Control on the Population Diversity

For every experiment, the diversity trace was recorded for each algorithm. Figure 1 shows 6 sub-plots of the average diversity change in the first 10 iterations for each problem instance. These graphs reveal that in all algorithms the diversity converges in the initial generations. In all instances of scenarios A and B there was an apparent change in the diversity value while using GADP, RCGADP and RCAGADP. This is a result of the disturbance caused by the dynamic population resizing which increases the number of dissimilar individuals. A closer inspection of the diversity values in generations 100 to 400 is displayed in Figure 2.

For instances in scenarios A, B and C, high initial value of the mutation rate in all methods delayed convergence in GA as it can be seen in the top plots of Figure 2. When GA, GADP and RCGADP were used, higher diversity was maintained in later generations. This indicates the merit of the dynamic population re-sizing with the help of crossovers exchange. On the other hand, algorithm variations with adaptive rates only, i.e. AGA and RCAGA, exhibited loss of diversity in sets A, B and C. This indicates that the adaptive rates require more time to work. This was corroborated for the larger instances in sets D, E and F, in which adaptive rates variations excelled in comparison to dynamic population re-sizing variations. The adaptive rates also enhanced the search by increasing population diversity even though a low mutation rate was set initially.

The bottom plots in Figure 2 show that there is a high level of intensification in the later generations. However, when random exchanges of crossover were used (RCAGA and RCAGADP), they promoted diversity values at all stages of the search. Henceforth, adaptive rates variations were able to enforce a change in the diversity when given more computational time. Thus, dynamic population re-sizing methods were capable of adapting quickly to environmental change only in smaller instances. On the other hand, dynamic re-sizing succeeded in increasing diversity in RCAGADP, but GADP and RCGADP had a loss in the diversity as a result of the low mutation rate used. Thus, it can be argued that the maintenance of mutation rates encourages the diversity further in larger problem sets along with the random crossover exchange.

C. Statistical Analysis of Adaptive Control Methods

To establish statistical significant in EAs, [23] recommended the use of Friedman analysis as a non-parametric statistical test to compare and rank methods. Here, an IBM SPSS 22 two-way analysis was used to compare the variances of seven related-samples, with a significant level of $\alpha = 0.05$ and 95% as a confidence interval. Table III provides the results generated by the Friedman analysis on the 42 problem instances including: the mean value, the standard deviation, the minimum cost-value, the maximum cost-value and the mean rank. The results presented in the mean rank row show the methods order based on the statistical analysis, where a low rank value indicates a better method and a high rank indicates a worse method. Four additional values were calculated and used to measure the performance of each algorithms **Gap%** is the average gap percentage to the best-known value, **Dev.** is the average percentage deviation from the best-known value (best solution of all the algorithms applied), **Best** is the fraction of instances in a set for which an algorithm matches the best-known value (best solution of all the algorithms applied). Additionally, the **Score** is the fraction of the instances for which a competing algorithm 'wins', i.e. produces better solutions than the configuration being scored. This score is calculated as $((q \times (p - 1)) - r)/(q \times (p - 1))$, where p is the number of methods compared, q is the number of problem instances, and r is the number of instances in which the $p - 1$ competing configurations find a better result. Hence, the best score value is 1, when $r = 0$, and the worst score value is 0, when $r = q \times (p - 1)$.

The results in Table III, have a significance level = 0.0, degrees of freedom = 6 and χ^2 = 55.507. This indicates

---

[1] The instances data and weights used here (blue setting) are available at https://drive.google.com/open?id=0B2OtHr1VocuSNGVOT2VSYmp6a2M.
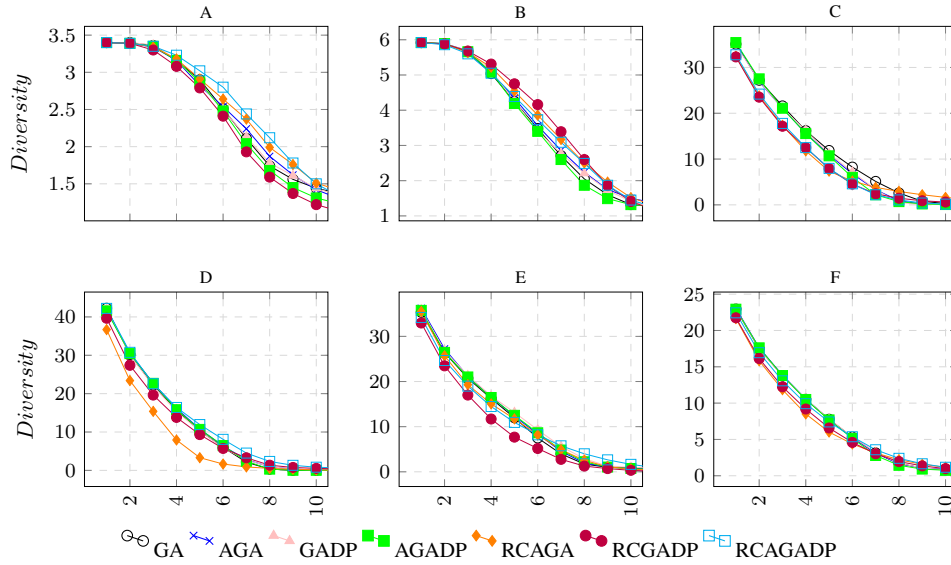
Fig. 1. Change in the diversity values in the first 10 iterations. Each line in the sub-plot represents a GA variation as follows. The GA (black), AGA (blue), AGADP (pink), GADP (green), RCAGA (orange), RCGADP (red) and RCAGADP (cyan).
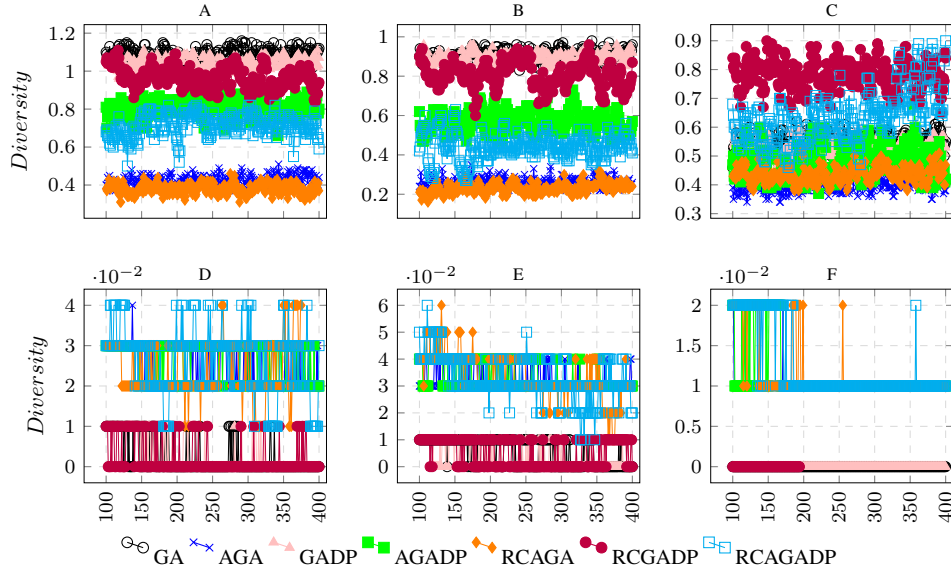

Fig. 2. Change in the diversity values in iterations 100-400. Each line in the sub-plot represents a GA variation as follows. The GA (black), AGA (blue), AGADP (pink), GADP (green), RCAGA (orange), RCGADP (red) and RCAGADP (cyan).

Table III. NON-PARAMETRIC FRIEDMAN'S TEST COMBINED WITH PERFORMANCES METRICS RESULTS

Method	IGA	AGA	GADP	AGADP	RCAGA	RCGADP	RCAGADP
Mean	332.49	289.00	292.40	289.95	288.23	307.59	**287.97**
STD Deviation	785.87	696.68	705.05	699.05	692.57	750.52	**691.86**
min	1.42	1.18	1.18	1.18	1.18	1.19	1.21
max	3980.44	3495.99	3497.99	3496.18	3500.00	3649.19	3496.18
Mean Rank	5.71	**3.25**	3.57	3.26	3.87	4.37	3.96
Gap%	0.16	**0.00**	0.00	0.00	0.01	0.02	0.01
Dev %	39.97%	**0.36%**	0.50%	0.39%	0.72%	1.71%	0.66%
# Best	0.19	**0.52**	0.36	0.38	0.33	0.36	0.33
Score	0.25	**0.75**	0.65	0.73	0.60	0.54	0.62

that there is clear evidence of statistically significance differences between all methods. All algorithms applied have obtained similar lower bounds except for IGA, RCGADP and RCAGADP. This outcome further shows that there is loss of performance in the uniform crossover exchange. The results for AGA was promising by obtaining the lowest *mean rank* and the highest *score*. While RCAGADP obtained the lowest *mean value* and *standard deviation*. These results indicate that the dynamic population re-sizing and the random crossover exchange are more effective for WSRP scenarios only when adaptive rates are applied. The AGA is the best method statistically because of its fast response to any environmental change, therefore it finds better solutions more frequently than other adaptive methods. To examine where the differences actually occurs, an additional analysis was implemented. The Wilcoxon signed test is used to rank the different methods based on their differences independently. Therefore, this test was influenced by the number of instances. The Bonferroni adjustment was applied on the results of the previous table to be able to preform multiple comparisons. The new significance level was

$0.05/7 = 0.007$, where 7 is the number of the methods compared. This means that if the $p < 0.007$; the distribution of a method i is different across the values of a method j, henceforth rejecting the null hypothesise by being statistically significant. If $p > 0.007$; the distribution of a method i is the same across values of a method j, therefore retaining the null hypothesise by having no statistical significance.

Table IV. NON-PARAMETRIC WILCOXON SIGN RANK TEST RESULTS

METHOD	Z	Sig. (2 tailed)	Sign	METHOD	Z	Sig. (2 tailed)	Sign
AGA - IGA	-4.64	0.00	S+	AGADP - GADP	-2.35	0.02	S-
GADP - IGA	-4.66	0.00	S+	RCAGA - GADP	-1.72	0.08	S+
AGADP - IGA	-4.60	0.00	S+	RCGADP - GADP	-3.48	0.00	S-
RCAGA - IGA	-3.90	0.00	S+	RCAGADP - GADP	-0.28	0.78	S+
RCGADP - IGA	-2.86	0.00	S+	RCAGA - AGADP	-1.46	0.14	S+
RCAGADP - IGA	-3.37	0.00	S+	RCGADP - AGADP	-3.47	0.00	S-
GADP - AGA	-2.68	0.01	S-	RCAGADP - AGADP	-1.44	0.15	S+
AGADP - AGA	-0.60	0.55	S-	RCGADP - RCAGA	-2.12	0.03	S-
RCAGA - AGA	-2.28	0.02	S+	RCAGADP - RCAGA	-0.19	0.85	S+
RCGADP - AGA	-3.35	0.00	S-	RCAGADP - RCGADP	-1.15	0.25	S+
RCAGADP - AGA	-2.26	0.02	S+				

Table IV shows the Wilcoxon signed-rank test. Column Z indicates the z-ratio as the unit for the normal distribution. Column Sig. (2-tailed) indicates the p value. Column sign indicates the statistical comparison of the null hypothesise (h_0) that denotes the significant difference between two methods. The S+ (S-) signs means that a method i was significantly better (worse) than a method j, within a confidence interval of 95%. In comparing the two related methods, the GA was worse than all other methods. This outcome further proves the superiority of adaptive methods over the non-adaptive GA for WSRP. The Wilcoxon signed-rank test showed that AGA, GADP and AGADP have a statistically significant advantage over RCGADP with (Z = -3.353, p = 0.0008), (Z = -3.476, p = 0.0005) and (Z = -3.473, p = 0.0005) respectively and no statistically significance difference across other methods. On the other hand, AGA, GADP and AGADP are worse than RCAGA and RCAGADP while RCAGA was worse than RCAGADP with no significant difference. These results statically prove that adaptive rates preforms the best when combined with a full adaptive method for WSRP scenarios.

D. Performance Comparison of Adaptive Control Methods

Table V presents a comparison between the results of the best adaptive algorithm variations (AGA and RCAGADP) against results obtained by the MIP method with decomposition in [7], the VNS in [8] and the GA in [5]. For each problem instance, the table shows the cost-value $f(S)$ and the computation-time in seconds *Cpt* in which a solution by a method was found. The best results are highlighted in **bold**.

In respect of the total number of best solutions obtained by each method, VNS and AGA performed the best producing best results on 45.24% and 35.71% of all instances respectively. In the set A, a full adaptive method obtained better results than AGA. In the set B, better results that those produced by GA were achieved by adaptive methods, but still not better than the VNS results. A similar observation can be made for set C. The adaptive mechanisms seem to have a positive effect on tackling instances in sets D, E and F where a number of best-known values were found by the adaptove GAs. In terms of computation time, RCAGADP produced good results with fast convergence on the E instances. AGA performed the best on sets D and F, arguably by enhancing the search through the adaptive mechanism. Overall, VNS found more best-known results but the GAs outperform the MIP decomposition method in terms of solution cost. It should be acknowledged that for several of the small instances the GAs spend considerably more time than the other methods, although still below 15 minutes which can be considered acceptable in practice. In general, the MIP decomposition method obtains good solutions in short computation-times across all sets of instances.

VI. CONCLUSIONS

Tuning parameter values for each problem instance can work well particularly in the initial stages of the evolutionary search. However, adjusting parameter values is known to be required for shifting the focus of the search on different areas [6]. In general, the goal of an adaptive parameter control method is to change the parameter settings during the search and only when necessary. In this paper, the aim was to investigate the effect of using a diversity-based adaptive parameter control technique for enhancing the performance of a genetic algorithm (GA) tailored for tackling the workforce scheduling and routing problem (WSRP). Three elements were included in the adaptive parameter control method: adaptive mutation/crossover rates, dynamic population size and random crossover exchange. As a result, six GA variations were compared in this paper in which the adaptive control is triggered by stagnation of the search. The (not-adaptive) GA produced poor results in comparison to the GA variants incorporating some adaptive mechanism.

Experimental results from solving a set of WSRP instances with the adaptive GA variants indicate that having such adaptive mechanisms is beneficial. Adapting the genetic operator rates produced better results especially for the larger WSRP scenarios. It is argued that the reason for this is that adaptive rates are adjusted faster according to any change in the environment, and therefore avoiding premature convergence of the search. More research is needed on the crossover selection in order to improve the algorithm efficiency by achieving an effective balance between solutions diversity and crossover selection.

REFERENCES

[1] R. Lassaigne and M. De Rougemont, *Logic and complexity*. Springer Science & Business Media, 2012.

[2] P. Cowling, N. Colledge, K. Dahal, and S. Remde, "The trade off between diversity and quality for multi-objective workforce scheduling," in *Evolutionary Computation in Combinatorial Optimization*, ser. Lecture Notes in Computer Science, J. Gottlieb and G. Raidl, Eds. Springer Berlin Heidelberg, 2006, vol. 3906, pp. 13–24.

[3] M. Mutingi and C. Mbohwa, "Health-care staff scheduling in a fuzzy environment: A fuzzy genetic algorithm approach," in *Conference proceedings (DFC Quality and Operations Management)*, 2014.

[4] H. ALgethami, D. Landa-Silva, and A. Martinez-Gavara, "Selecting genetic operators to maximise preference satisfaction in workforce scheduling and routing problem," in *Proceedings of the 6th International Conference on Operations Research and Enterprise Systems (ICORES).*, Porto, Portugal, 2017, pp. 416–423.

[5] H. Algethami, R. L. Pinheiro, and D. Landa-Silva, "A genetic algorithm for a workforce scheduling and routing problem," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 927–934.

[6] A. E. Eiben, M. Horvath, W. Kowalczyk, and M. C. Schut, "Reinforcement learning for online control of evolutionary algorithms," in *Engineering Self-Organising Systems: 4th International Workshop, ESOA 2006, Hakodate, Japan, May 9, 2006, Revised and Invited Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 151–160.

Table V.   RESULTS OF COST-VALUES ($f(S)$) AND COMPUTATIONAL-TIMES (Cpt) FOR ALL SOLUTION METHODS ON 42 PROBLEM INSTANCES.

		Optimal		MIP DECOMP.		VNS		GA		AGA		RCAGADP	
		$f(s)$	Cpt	$f(s)$	Cpt	$f(s)$	Cpt	$f(s)$	Cpt	$f(s)$	Cpt	$f(s)$	Cpt
A	1	3.49	7.00	5.65	3.71	3.49	2.99	3.49	34.08	3.49	14.45	**3.49**	**1.82**
	2	2.49	8.00	4.53	3.58	**2.49**	**0.33**	2.96	396.77	2.49	100.69	2.49	3.09
	3	**3.00**	**14.00**	10.65	3.70	3.00	29.30	3.65	436.92	3.08	11.50	3.02	206.43
	4	**1.42**	**5.00**	3.09	2.88	1.42	6.87	1.42	229.01	1.42	247.07	1.42	81.06
	5	2.42	1.00	3.54	1.77	**2.42**	**0.11**	2.42	0.98	2.42	1.11	2.42	0.58
	6	**3.55**	**5.00**	3.74	2.42	3.55	29.47	3.61	159.83	3.56	130.22	3.55	24.70
	7	3.71	1.00	4.81	1.64	**3.71**	**0.07**	3.71	0.83	3.71	0.72	3.71	0.65
B	1	**1.70**	**21.00**	1.79	8.07	1.70	156.45	1.76	422.62	1.75	84.00	1.74	274.61
	2	1.75	2.00	1.89	4.29	**1.75**	**0.00**	1.75	78.57	1.75	0.56	1.75	0.95
	3	1.72	6003.00	2.06	32.86	**1.72**	**473.21**	1.90	629.30	1.82	280.35	1.80	229.30
	4	2.07	25.00	2.21	15.25	**2.07**	**9.03**	2.08	29.51	2.08	46.19	2.08	4.08
	5	**1.82**	**585.00**	4.74	25.35	1.83	135.33	1.96	434.63	1.93	98.85	1.92	266.73
	6	**1.62**	**184.00**	2.52	24.11	1.62	194.29	1.71	343.74	1.66	190.74	1.65	244.94
	7	**1.79**	**300.00**	4.06	23.64	1.79	304.96	1.87	502.96	1.81	154.78	1.81	239.79
C	1	n/a	n/a	904.98	211.64	**114.21**	**301.32**	115.13	846.87	115.07	703.71	116.77	812.12
	2	3.15	6.00	3.61	0.57	**3.15**	**0.00**	3.15	0.46	3.15	0.46	3.15	0.50
	3	n/a	n/a	1186.27	26.33	**103.52**	**550.29**	104.83	657.11	105.08	677.45	104.97	698.98
	4	11.15	90.00	81.25	3.09	**11.15**	**0.91**	11.15	2.41	11.15	2.40	11.15	2.88
	5	12.34	55.00	68.94	1.05	**12.34**	**0.34**	12.34	3.38	12.60	33.70	12.41	177.26
	6	n/a	n/a	3102.26	47.05	**140.44**	**323.52**	180.98	591.22	180.99	589.30	180.93	800.68
	7	4.30	1.00	5.29	0.24	4.30	0.02	**4.30**	**0.00**	**4.30**	**0.00**	**4.30**	**0.00**
D	1	n/a	n/a	496.45	1059.99	170.33	3036.23	171.34	2789.97	170.77	2978.51	**170.77**	**1663.00**
	2	n/a	n/a	372.94	1192.05	**163.85**	**2840.76**	176.68	3600.00	166.87	3228.52	176.98	3837.26
	3	n/a	n/a	3213.32	1208.99	178.19	3172.79	181.56	2375.40	**180.60**	**3258.30**	181.87	2656.32
	4	n/a	n/a	418.89	3005.14	**167.11**	**3313.84**	172.83	2881.37	167.93	3465.12	169.13	3576.84
	5	n/a	n/a	243.89	1306.70	161.11	2818.51	161.81	2084.21	**161.73**	**3367.04**	165.14	3105.46
	6	n/a	n/a	1411.27	1222.00	**177.43**	**2995.95**	178.54	2630.91	178.28	3200.96	178.81	3081.79
	7	n/a	n/a	753.28	1361.89	177.86	2930.41	178.85	2391.55	**178.40**	**2489.11**	178.92	5587.61
E	1	n/a	n/a	33.01	8408.01	n/a	n/a	**1.18**	**3210.88**	1.18	3322.13	1.21	3426.49
	2	n/a	n/a	26.05	12448.42	n/a	n/a	1.27	3615.84	1.21	3368.60	**1.21**	**1258.55**
	3	n/a	n/a	28.99	20746.58	n/a	n/a	**1.21**	**2227.97**	1.21	3246.99	1.26	3292.24
	4	n/a	n/a	28.52	15190.50	n/a	n/a	1.30	3445.58	**1.29**	**3492.21**	1.30	2665.07
	5	n/a	n/a	270.09	32619.21	n/a	n/a	2.29	3592.91	2.24	3442.02	**2.24**	**2440.54**
	6	n/a	n/a	24.63	24212.11	n/a	n/a	1.30	3062.06	**1.30**	**3283.59**	1.30	2898.00
	7	n/a	n/a	427.78	51057.29	n/a	n/a	1.73	2044.52	1.73	3473.85	**1.73**	**2030.66**
F	1	n/a	n/a	64305.09	3446.42	n/a	n/a	**2029.05**	28433.97	2122.60	27647.63	2036.26	19988.47
	2	n/a	n/a	73291.22	1111.05	n/a	n/a	2092.49	20093.64	**2082.79**	**26318.27**	2088.17	24745.08
	3	n/a	n/a	115235.17	4555.11	n/a	n/a	609.21	19776.63	**607.26**	**27722.40**	619.77	28488.47
	4	n/a	n/a	102994.17	4219.20	n/a	n/a	1342.16	23617.59	**1328.51**	**27229.31**	1329.46	14089.44
	5	n/a	n/a	101438.18	6156.54	n/a	n/a	198.90	24010.24	**196.08**	**26934.63**	198.08	13129.58
	6	n/a	n/a	76007.08	9695.61	n/a	n/a	637.16	25892.93	**624.77**	**25367.39**	628.41	12428.78
	7	n/a	n/a	176540.59	3832.77	n/a	n/a	3500.00	15089.69	**3495.99**	**27225.77**	3496.18	7450.12

Bold text refers to the best result.

[7] W. Laesanklang and D. Landa-Silva, "Decomposition techniques with mixed integer programming and heuristics for home healthcare planning," *Annals of Operations Research*, pp. 1–35, 2016.

[8] R. L. Pinheiro, D. Landa-Silva, and J. Atkin, "A variable neighbourhood search for the workforce scheduling and routing problem," *Proceedings of the 7th World Congress on Nature and Biologically Inspired Computing (NaBIC 2015)*, December 2015.

[9] H. Algethami and D. Landa-Silva, "A study of genetic operators for the workforce scheduling and routing problem," *MIC 2015 conference proceedings*, 2015.

[10] S. van Rijn, M. Emmerich, E. Reehuis, and T. B?ck, "Optimizing highly constrained truck loadings using a self-adaptive genetic algorithm," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, pp. 227–234.

[11] B. M. Ginley, J. Maher, C. O'Riordan, and F. Morgan, "Maintaining healthy population diversity using adaptive crossover, mutation, and selection," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 692–714, Oct 2011.

[12] D. Goldberg, K. Deb, and B. Korb, "Messy genetic algorithms: motivation, analysis, and first results," *Complex Systems*, pp. 493–530, 1989.

[13] R. E. Smith and E. Smuda, "Adaptively resizing populations: Algorithm, analysis, and first results," *Complex Systems*, vol. 9, pp. 47 – 72, 1993.

[14] D. Mankowska, F. Meisel, and C. Bierwirth, "The home health care routing and scheduling problem with interdependent services," *Health Care Management Science*, vol. 17, no. 1, pp. 15–30, 2014.

[15] Y.-Y. Wong, K.-H. Lee, K.-S. Leung, and C.-W. Ho, "A novel ap-

proach in parameter adaptation and diversity maintenance for genetic algorithms," *Soft Computing*, vol. 7, no. 8, pp. 506–515, 2003.

[16] W. M. Spears, "Adapting crossover in evolutionary algorithms," in *Proceedings of the Fourth Annual Conference on Evolutionary Programming*. MIT Press, 1995, pp. 367–384.

[17] E. Osaba, E. Onieva, R. Carballedo, F. Diaz, and A. Perallos, "An adaptive multi-crossover population algorithm for solving routing problems," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2013)*. Springer, 2014, pp. 113–124.

[18] R. W. Morrison and K. A. De Jong, "Measurement of population diversity," in *Artificial Evolution*. Springer, 2002, pp. 31–41.

[19] J. D. Landa Silva, "Metaheuristic and multiobjective approaches for space allocation," Ph.D. dissertation, University of Nottingham, 2003.

[20] Z. S. Ma and A. W. Krings, "Dynamic populations in genetic algorithms," in *Proceedings of the 2008 ACM symposium on Applied computing*. ACM, 2008, pp. 1807–1811.

[21] F. G. Lobo and C. F. Lima, "Adaptive population sizing schemes in genetic algorithms," in *Parameter Setting in Evolutionary Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 185–204.

[22] S. Hartmann, "A competitive genetic algorithm for resource-constrained project scheduling," *Naval Research Logistics (NRL)*, vol. 45, no. 7, pp. 733–750, 1998.

[23] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization," *Journal of Heuristics*, vol. 15, no. 6, p. 617, 2008.