Effectiveness of Selection Mechanisms on the efficiency of Multi Parent Crossover Operator

1 author:

Esam Taha Yassen
University of Anbar

ISSN: 1991-8941

Effectiveness of Selection Mechanisms on the efficiency of Multi Parent Crossover Operator

Esam Taha Yassen
University of Anbar- College of Computer Science and Information Technology.

Abstract: Multi-parent crossover has been proven its ability to address many of combinatorial optimization problems such as the traveling salesman problem and the vehicle routing problem with time windows. The successful use of multi-parent crossover arises from its abilities to enhance the search performance via utilizing information exchanged by more than two parents and inheriting by offspring. These parents are selected according to one of the selection mechanisms. Selecting the most appropriate parents for a crossover process might leads to improving the effectiveness of genetic algorithm. Therefore, this work investigates the effect of selection mechanism on the efficiency of multi-parent crossover. To test this, seven selection mechanisms have been used; random selection mechanism, roulette wheel mechanism, stochastic universal sampling mechanism, tournament selection mechanism, best selection mechanism, single best-couple random selection mechanism and couple best-single random selection mechanism. The performance of the proposed algorithm is tested using Solomon VRPTW benchmark. The experimental results show the superiority of multi-parent crossover that employs the selection mechanism which selects the outstanding individuals to form most of parents over multi-parent crossover that employ other selection mechanisms. This demonstrates the efficiency of employing best parents in a crossover process that can assist the search process to attain a better solution.

Keywords: Vehicle routing problem; Genetic algorithm; Selection mechanism; Crossover

1. INTRODUCTION

Vehicle routing problem with time windows (VRPTW) described as one of the challenging combinatorial optimization problems [1]. VRPTW is an extended version of traditional Vehicle routing problem [2] that considers time restriction [3, 4]. The main aim of resolving the VRPTW is to design a group of vehicle routes that can serve a group of customers with the least cost and avoid any violation to the prescribed restrictions. Four obligatory restrictions must be regarded; these are: (1) a vehicle must begin and stop at the depot, (2) the overall requests of customers which are allocated to a vehicle must not exceed the capacity of the vehicle, (3) each customer must be served within the required time window, (4) split deliveries should be avoided. The goal of solving VRPTW is to generate feasible routes to serve all customers with minimal cost (see Equation (1)). Assume the following variables [5] :

$$x_{ij}^{k} = \begin{cases} 1 & \text{if vehicle } k \text{ travelled directly from customer } c_i \text{ to } c_j \\ 0 & \text{Otherwise} \end{cases}$$

v *number of serving vehicles.*
n *number of customer nodes (excluding the depot node).*
t_{ij} *travel time from customer c_i to customer c_j.*

The quality of the solution S is measured using an objective function:

$$f(S) = \min \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{v} t_{ij} \times x_{ij}^{k} \tag{1}$$

Due to the fact that exact approaches work effectively only in solving NP-hard (Non-Deterministic Polynomial-Time Hard) problems with a small size, meta-heuristic procedures are usually desirable as solution techniques to address NP-hard problems [5]. Consequently, there are a great number of meta-heuristic algorithms that have been employed to deal with the VRPTW as an NP-hard problem [6]. In

1995, a tabu search algorithm with a probable strategy was introduced by Rochat and Taillard to create a balance between diversification and intensification [7]. Russell and Chiang employed a scatter search to solve the VRPTW [8]. Nazif and Lee suggested a genetic algorithm that is equipped with an optimized crossover factor to deal with the VRPTW [9]. Wang hybridized an ant colony algorithm with iterated local search algorithm so as to avoid convergence with the local optima [10]. [5] utilized a set-based particle swarm optimization for solving VRPTW.

Genetic algorithm (GA) was proposed by [11]. Since then GA has been popular, because it can contribute to detect good solutions for complex mathematical problems in a reasonable amount of time [12, 13]. GA is unlike the classical local search algorithm in that it is able to operate on a big group (population) of solutions. The core operator of GA is the crossover operator which is employed to merge two solutions (parents) to swap the information from one solution to another. In terms of the crossover or the combination operator, some types of crossover operators were presented during the past years including one point and multi points crossovers [14-17]. It is argued that not all types of crossover operators might be suitable to deal with vehicle routing problem. This is because of the restrictions which have to be taken care of to avoid having infeasible solutions [13, 18]. Accordingly, several types of crossover operators have been designed to deal with vehicle routing problem like the order crossover (OX) [19] and edge-assembly crossover (EAX) [20].

In spite of the efficacy of two parent crossover, recent works demonstrate that the efficacy of GA is revised via utilizing a crossover of multi parent instead of two parents [21-23]. The use of more than two parents in the crossover process leads to producing offspring who carry new various features as they inherit their features from different parents. Accordingly, the multi-parent crossover, in addition to the mutation operator, increases the exploration of the search at the expense of the search exploitation. This will negatively affect the efficiency of the search and leads to obtain less competitive solutions. In GA, the selection process is responsible for choosing the parents for mating and guiding the search to the good region within the search space (where the best solutions may be located). So, The parent's selection mechanism has its clear effect on attaining the exploration-exploitation balance which has a positive effect on the search efficiency [13, 18].

Different selection mechanisms have been used in previous works such as the roulette-wheel mechanism [24, 25], tournament selection mechanism [26], [27], [28], [29], [30] and [31], rank-based selection mechanism [32] and [33] and random selection mechanism was adopted by some authors like [34-37]. However, at this point an essential question will be raised:

Which of these selection mechanisms is better and why?

This work aims to investigate the effectiveness of seven selection mechanisms on multi parent crossover. These are: random selection mechanism, roulette wheel mechanism, stochastic universal sampling mechanism, Tournament selection mechanism, best selection mechanism and two mixed selection mechanisms. Experiments are conducted on the Solomon's VRPTW benchmark.

2. GENETIC ALGORITHM

GA was proposed by Holland in 1975 [11]. Since then, GA has been popular, because it can solve various optimization problems, like the VRP, in a reasonable amount of time [5] [18]. GA evolves a population of solutions while searching for a better quality solution by using an evolutionary process. The major procedures within GA include the selection, crossover, mutation and updating processes [38]. GA begins with a group of solutions created either arbitrarily or via a specific version of heuristic algorithms. Then, the fitness of these solutions is calculated. After that the process of selection is employed to choose two solutions known as the parents, based on the fitness value [39]. These parents will pass through the crossover and/or mutation operator. The goal of the crossover operator is to merge the chosen parents so as to utilize or interchange information between these parents to create the offspring. The mutation operator can obtain the necessary diversity and protect the search against getting stuck in local optima. When the offspring is more efficient than the worst one in the present population, it will replace the worst solution. This procedure is reiterated several times and the process is called the generation in GA (see algorithm 1) [38]:

Algorithm 1. A typical Genetic Algorithm

```
Start
    P ← initial population;
    Evaluate (P);
    While termination criterion not satisfied Do
        P' ← recombines (selected (P));
        Mutate (P');
        Evaluate (P');
        P ← replace (P' ∪ P);
    End while
End
```

Commonly, the most significant operator in GA is the crossover operator due to its ability to manage the quantity of information needed to be swapped and inherited by the subsequent generation [9]. The role of crossover is to regulate the intensification process to direct the whole search process to a specific region. Many types of crossover operators have been presented including one point and multi points crossovers. Each crossover operator has certain drawbacks and only operate successfully to tackle some types of problems [14, 18]. Not all types of crossover operators can be utilized to deal with VRPTW. This is because of the restrictions that have to be taken care of to avoid having infeasible solutions [1, 18]. Accordingly, several types of crossover operators have been exactly planned to deal with VRP like the order crossover (OX) [19] and edge-assembly crossover (EAX) [20]. Numerous works that focused on utilizing multi parents crossover have been lately reported [21, 22, 40]. The key point behind applying multi parents crossover is to obtain more exploration [21] [22].

3. PROPOSED ALGORITHM

In this work, GA with multi-parent crossover (Multi-GA) is used to deal with VRPTW. In VRPTW, chromosome, gene and node; represent solution, route and customer respectively. Fig. 1 shows the flowchart of the Multi-GA which consists of the following steps:

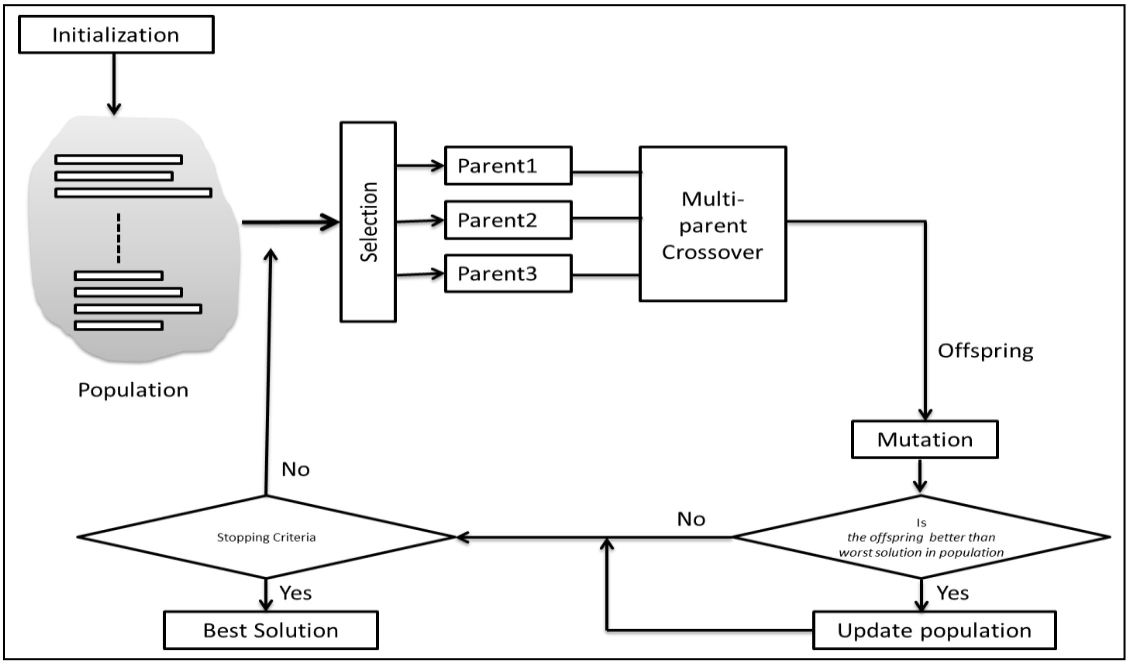

Fig. 1 Flow chart of Multi-GA.

a. **Initial population**: In this work, the population of VRPTW is initialized using the greedy constructive heuristic as follows: Firstly, an empty route is created. Then, a customer is randomly selected and added to the current route as long as it satisfies the imposed constraints. When no customer can be added to the current route, a new route is created. The process of creating new routes and inserting customers is reiterated till all customers are inserted to a particular route.

b. **Selection**: When the population is created, this process proceeds to select parents for mating. Lots of selection mechanisms have been used in previous works [1, 18, 41]. Seven of these mechanisms are adopted in this work to investigate the effectiveness of each selection mechanism on the performance of Multi-GA:

- Random Selection mechanism (RSM): in this mechanism, parents are selected randomly from the population regardless of its quality.
- Best Selection mechanism (BSM): in this mechanism, best individuals (whose fitness value is the best) are adopted to be the parents.
- Mixed Selection mechanism: in order to investigate the effect of involving the best parents and random parents in mating process, this work proposed a mixed selection mechanism. This mechanism represents the operation of mixing both early mentioned mechanisms, where some parents represent the best individuals and other parents will be selected randomly. On the base of the parents selection, mixed selection mechanism will be classified into two selection mechanisms:
-
- Single Best-Couple Random Selection Mechanism (SBCR): In this mechanism, the best individual will be selected to be the first

parent and the other parents will be randomly selected.

- Couple Best- Single Random Selection Mechanism (CBSR): In this mechanism, the two best individuals will be selected as first and second parents and the third parent will be randomly selected.
- Best-Random Selection mechanism (BRSM): in this mechanism, all of parents represent the best individuals (whose fitness values are the best) in population except one that is randomly selected.
- Roulette Wheel Selection mechanism (RWSM): The roulette wheel is the most widely used selection mechanism [42, 43] which is proposed by [44] . In this mechanism, each individual within the population is assigned to its selection probability (based on its fitness) and all parents are selected based on their selection probabilities, (see Fig. 2).

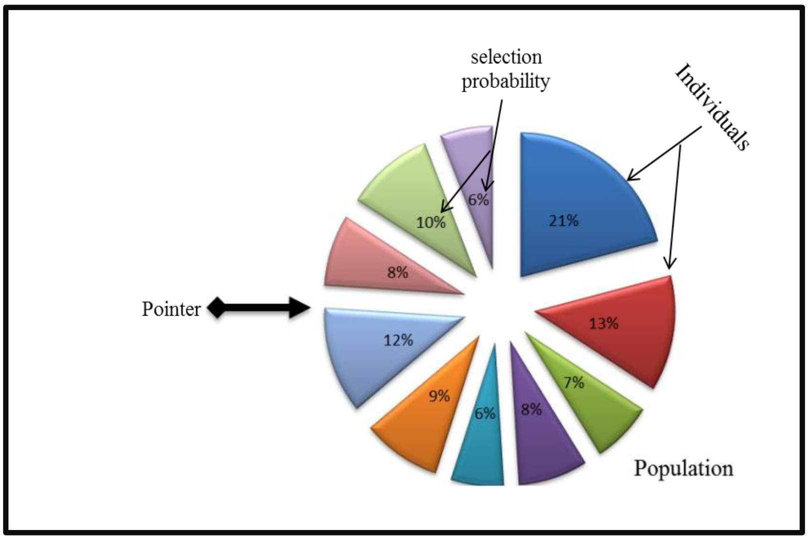

Fig. 2 The roulette wheel selection mechanism

- Stochastic Universal Sampling mechanism (SUSM): Instead of only one pointer is employed to select the individual as in RWSM, this mechanism utilizes number of similarly spaced pointers according to the number of selections required (N). The distance between these pointers is equal (1/N) and the place of the first pointer is randomly generated [13, 18]. Fig. 3 illustrates this selection process.

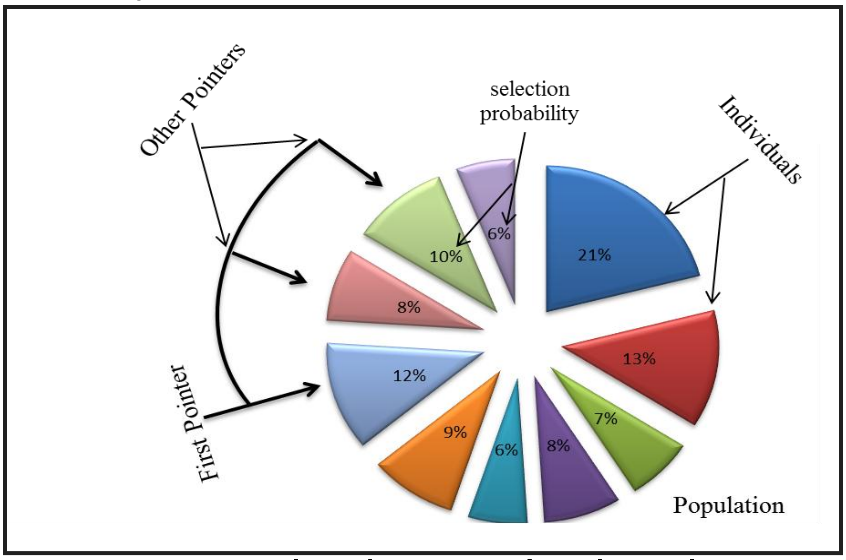

Fig. 3 The stochastic universal sampling mechanism

- Tournament Selection mechanism (TSM): In this mechanism, k individuals are randomly selected to form a tournament group of k size. Then, the best individual within the tournament group will be selected as a one parent [13, 18, 45]. The rest of parents will be selected in the same process. The procedure of this mechanism is illustrated in Fig. 4.

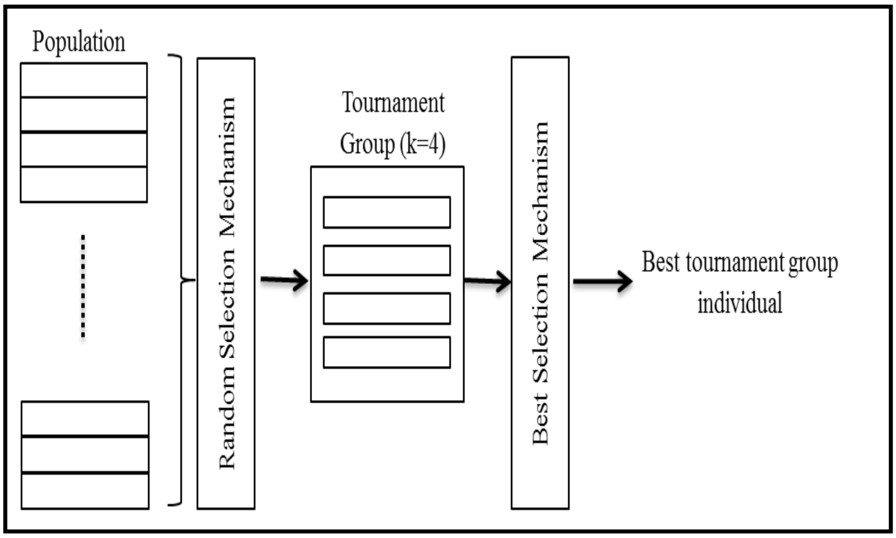

Fig. 4 The tournament Selection mechanism

c. **Recombination:** This paper aims to develop an insertion crossover of two parents [38] into a multi parent. The process of the suggested multi parent insertion crossover can be described as follows: Each parent is assigned to its inheritance probability (*IP*) which is calculated by Equation 2 [18]:

$$IP_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \qquad \forall i = 1..n \tag{2}$$

Where n is the number of parents and f_i is the quality of parent *i*. Thereafter, the offspring inherits its attributes (genes) from its parents on the base of the parents' *IPs*.

Since VRPTW is a constrained problem, generated offspring is usually infeasible because some customers are either missed or duplicated. To rectify the infeasible offspring, a repair mechanism is applied. The following steps illustrate the general procedure of this mechanism:

1- Determine the duplicated and missed customers.

2- Remove all duplicated customers from the routes.

3- For each missed customer, do the following:

- Try to allocate it to any possible route (if it does not violate the constraints).
- If the customer cannot be located to any possible route, a new route will be created for it.

An example of three parent insertion crossover is illustrated in Fig. 5.

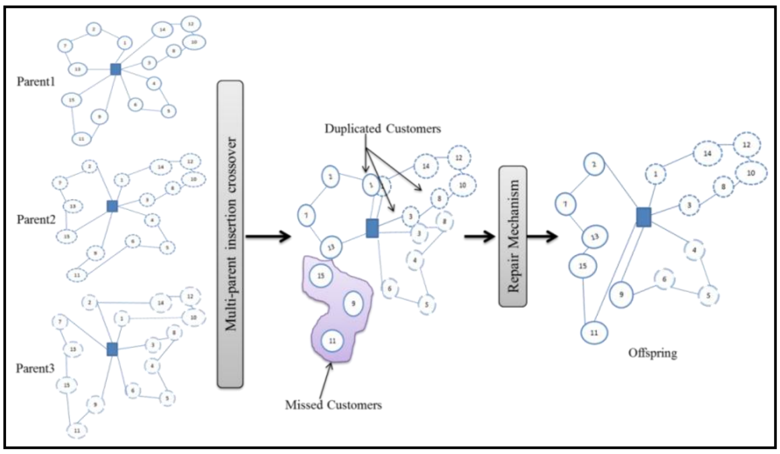

Fig. 5 An example of three parent insertion crossover

d. Mutation: Two-opt star operator is used in this step to generate the neighborhood solution [1]. This operator works as follow: select two routes randomly from the offspring and try to exchange the customers located at the end sections of these routes.

e. Evaluation: We use equation (1) to evaluate the fitness value of each one of the chromosomes.

f. Replacement: When the neighborhood solution (offspring) is more efficient than the poorest individual in the present population, it will substituted by the poorest individual. If not, the newly created neighborhood solution will be rejected.

g. Termination process: This phase examines the GA termination criterion. If the termination criterion is fulfilled GA will halt and give back the best obtained solution. If not, the steps b-f are reiterated.

Algorithm 2 shows the main components of the Multi-GA which utilizes the CBSR selection mechanism to select parents for mating.

Algorithm 2 The proposed Multi-GA-CBSR algorithm

Start
P $\leftarrow$ initial population;
Evaluate (P);
While termination criterion not satisfied **Do**
/* Selection */
The two best individuals are selected as 1st & 2nd parents
The 3rd parent is randomly selected
/* Recombination */
Assign each parent to its inheritance probability (IP) using Equation 2
Create an empty **offspring**
The **offspring** inherits genes from its parents based on their IPs
/* Mutation */
Offspring $\leftarrow$ two-opt star (**offspring**);
Evaluate (**offspring**);
Update population p;
End while
End

4. EXPERIMENTAL DESIGN

Multi-GA is applied using Java and accomplished via Intel processor Quad CPU 2.33GHz, RAM 2.00 GB. Solomon's VRPTW benchmark [46, 47] has been utilized to assess the performance the proposed algorithms. This benchmark involves 56 instances which are categorized into three datasets based on the distribution of customer; which are:

1. R is randomly.
2. C is clustered.
3. RC mixed (randomly and clustered).

These three datasets are also grouping based on the customer service time windows where R1, C1 and RC1 involve customers who have short time windows, whilst R2, C2 and RC2 involve customers who have large time windows.

4.1 Parameters Setting

A preliminary test is conducted to determine the suitable parameters values. During the preliminary test, Multi-GA is executed 10 times on six instances (R1-01, R2-01, C1-01, C2-01, RC1-01 and RC2-01) and the best results are reported. Table 1 demonstrates the parameters setting used for Multi-GA.

Table 1 Parameters settings

Parameter	Value
Population size	30
Crossover probability	0.75
Number of iteration	1000

4.2 Experimented Results

To investigate the effectiveness of the proposed Multi-GA in solving the VRPTW, two types of experiments are adopted in these works. The first experiment (Section 4.2.1) aims to investigate the effect of the number of parents on the performance of the Multi-GA. The second experiment is designed to verify the effect of the selection mechanisms on the Multi-GA performance.

To show if the obtained results are statistically significant or not, a statistical test is employed as follows: initially, Shapiro-Wilk normality test with 0.05 critical level is conducted to verify whether the distribution of obtained results is normal or not normal. This test demonstrates that the obtained results are not normally distributed (the p-value is less than 0.05). Accordingly, non-parametric test (Friedman test) has been used to validate that the obtained results are statistically different [18, 48, 49].

4.2.1 Effect of the Number of Parents

Results in Table 2 show the effect of the number of parents on the Multi-GA's performance. Three number of parents are tested (3, 4 and 5) and the resultants are three different Multi-GAs denoted as Multi-GA1, Multi-GA2 and Multi-GA3, respectively. All these algorithms are executed 31 runs on six instances (R1-01, R2-01, C1-09, C2-06, RC1-01 and RC2-01) and the best obtained results are reported. For each tested instance, we report the best (*Best*), the average (*Avr*) and the standard deviation (*Std*). Based on the *Best* and the *Avr* values, this comparison illustrates that the Multi-GA1is better than others on five and six instances out of six, respectively. Regarding the *Std*, the best *Std* results are distributed among three algorithms (each of them obtains the best results on two instances out of six).

Table 2 Comparison between Multi-GA1, Multi-GA2 and Multi-GA3

Instances	Multi-GA1			Multi-GA2			Multi-GA3		
	Best	*Avr*	*Std*	*Best*	*Avr*	*Std*	*Best*	*Avr*	*Std*
R1-01	**2255.94**	**2425.08**	54	2359.07	2487.64	**49.5**	2293.11	2475.56	56.96
R2-01	**1872.51**	**2010.18**	77.57	1902.8	2066.03	81.11	1979.05	2113.09	**65.76**
C1-01	**2434.42**	**2621.01**	86.63	2449.62	2623.95	81.58	2538.70	2689.2	**69.49**
C2-06	**1316.12**	**1829.84**	182.3	1540.24	1926.42	**156.86**	1624.44	2011.75	163.62
RC1-01	**2439.15**	**2584.41**	**75**	2473.47	2614.92	79.90	2492.75	2650.73	88.86
RC2-01	2281.6	**2401.88**	**62.15**	2258.49	2460.27	84.42	**2224.9**	2464.17	104.59

Furthermore, Friedman test has been conducted to identify the ranking of these algorithms. As illustrated in Table 3, the Multi-GA1 is in the first rank, as it achieved the lowest value whereas, Multi-GA2 and Multi-GA3 are in the 2^{nd} and 3^{rd} ranks, respectively.

Table 3 The ranking of Multi-GA1, Multi-GA1and Multi-GA1

Algorithm	Ranking	Index
Multi-GA1	1	(1)
Multi-GA2	2.17	(2)
Multi-GA3	2.38	(3)
Friedman test(*p-value*)	0.005704	
Iman-Davenport(*p-value*)	0.000052	

It is worth noting that the *p-value* of the Friedman and Iman-Davenport statistical tests demonstrated that, there is a considerable difference between the results of Multi-GA1, Multi-GA2 and Multi-GA3 (p-value < 0.05, i.e. p-value=0.00). Table 4 shows that the Multi-GA1 (which is the controlling method since it was in the first rank) is statistically better than Multi-GA2 and Multi-GA3 with critical level of 0.05 (adjusted p-value < 0.05).

Table 4 The adjusted *p*-value

Multi-GA1 *vs*	*p-value*
Multi-GA2	0.0015
Multi-GA3	0.0433

Overall, the results demonstrated that utilizing three parents can effectively assist the Multi-GA to work well and then produce good results across all instances.

4.2.2 Effect of the Selection Mechanisms

The impact of the selection mechanisms on the Multi-GA performance is validated in this subsection. Seven selection mechanisms are adopted in this work: RSM, BSM, SBCR, CBSR, RWSM, SUSM and TSM. These mechanisms produced seven various Multi-GA algorithms named: Multi-GA-RS, Multi-GA-BS, Multi-GA-SBCR, Multi-GA- CBSR, Multi-GA-RW, Multi-GA-SUS and Multi-GA-TS, respectively. Twelve instances named R101, R102, R201, R202, C101, C109, C201, C206, RC101, RC102, RC201 and RC202 are chosen to test these algorithms and the results over 31 runs are reported. Tables 5-7 show the results of applying the seven selection mechanisms on Multi-GA.

best solution (*Best*), average solution (*Avr*) and standard deviation (*Std*) are reported.
According to the *Best* values, Table 5 demonstrates that the best results were distributed among the Multi-GAs which involve the best individuals in crossover process. Multi-GA-BR, Multi-GA-SBCR and Multi-GA- CBSR obtained the best results on seven, four and two instances out of twelve, respectively. Based on the *Avr* values, Table 7 shows that the Multi-GA-CBSR obtained the best results on ten instances out of twelve and Multi-GA-BS obtained the best *Avr* results on two instances. With regard to the *Std*, Table 8 illustrates that the algorithm Multi-GA-TS obtained the best results on ten instances out of twelve, whilst Multi-GA-RS and Multi-GA-SUS achieved the best results on the rest two instances, each of them obtained the best results on one instance.

Table 5 Comparison between Multi-GAs in terms of *Best*

Instances	***Best***						
	Multi-GA-RS	Multi-GA-BS	Multi-GA-SBCR	Multi-GA-CBSR	Multi-GA-RW	Multi-GA-SUS	Multi-GA-TS
R1-01	2255.94	2158.46	**2136.54**	2149.25	2268.78	2286.26	2360.87
R1-02	2210.48	2008.33	2026.07	**2000.92**	2136.77	2211.05	2272.31
R2-01	1872.51	1762.13	**1724.43**	1740.88	1936.96	1937.19	1966.29
R2-02	1610.54	1624.62	1578.56	**1461.81**	1682.85	1660.48	1815.52
C1-01	2170.47	**1618.74**	1720.56	1646.96	2261.65	2243.63	2441.65
C1-09	2434.42	**1901.92**	2061.75	1993.47	2423.46	2379.41	2654.51
C2-01	732.43	**591.56**	594.32	**591.56**	915.33	1479.28	1373.15
C2-06	1316.12	1082.16	921.55	**889.31**	1335.73	1479.28	1933.08
RC1-01	2439.15	**2145.60**	2192.04	2212.72	2357.48	2342.76	2479.63
RC1-02	2274.88	2009.84	2034.36	**1961.68**	2269.89	2249.05	2351.13
RC2-01	2281.60	2101.44	2142.26	**2039.41**	2167.71	2206.84	2378.66
RC2-02	1956.26	1828.09	1799.48	**1691.68**	1939.15	1865.32	2145.77

Table 6 Comparison between Multi-GAs in terms of *Avr*

Instances	***Avr***						
	Multi-GA-RS	Multi-GA-BS	Multi-GA-SBCR	Multi-GA-CBSR	Multi-GA-RW	Multi-GA-SUS	Multi-GA-TS
R1-01	2425.08	2312.41	2316.10	**2304.20**	2429.37	2415.94	2481.81
R1-02	2314.73	2135.84	2154.63	**2113.86**	2298.72	2297.12	2371.72
R2-01	2010.18	1987.54	1944.38	**1919.16**	2029.16	2035.96	2140.90
R2-02	1837.25	1804.64	1746.54	**1697.44**	1809.65	1841.58	1931.15
C1-01	2485.89	1983.74	2105.44	**1966.10**	2482.31	2467.69	2619.75
C1-09	2621.01	**2125.18**	2273.87	2217.38	2637.92	2627.42	2784.42
C2-01	1234.83	784.71	888.32	**765.64**	1246.11	1826.68	1710.55
C2-06	1829.84	1676.66	1433.23	**1407.83**	1844.67	1826.68	2139.68
RC1-01	2584.41	**2347.35**	2406.38	2367.91	2600.11	2579.92	2649.25
RC1-02	2434.71	2190.66	2221.87	**2169.00**	2414.50	2440.34	2566.53
RC2-01	2401.88	2273.59	2250.18	**2214.62**	2378.81	2392.49	2527.51
RC2-02	2100.56	2060.28	1994.88	**1935.87**	2114.53	2105.47	2259.41

Table 7 Comparison between Multi-GAs in terms of *Std*

Instances	***Std***						
	Multi-GA-RS	Multi-GA-BS	Multi-GA-SBCR	Multi-GA-CBSR	Multi-GA-RW	Multi-GA-SUS	Multi-GA-TS
R1-01	54.00	71.21	79.20	94.92	48.03	57.96	**47.80**
R1-02	48.25	81.49	67.82	63.01	53.23	52.26	**47.43**
R2-01	77.57	80.43	80.24	98.00	72.01	**56.48**	58.23
R2-02	62.47	100.98	88.62	88.25	80.59	67.25	**59.92**
C1-01	119.77	205.81	194.67	158.77	109.38	102.23	**73.86**
C1-09	86.63	108.92	112.20	121.80	99.29	98.26	**67.92**
C2-01	211.83	321.12	258.93	277.96	180.68	168.56	**138.27**
C2-06	182.30	258.33	267.70	232.86	171.37	168.56	**118.52**
RC1-01	75.00	93.88	82.65	114.53	94.88	92.75	**69.70**
RC1-02	83.67	105.60	115.97	120.22	67.21	94.66	**67.73**
RC2-01	**62.15**	102.69	81.88	94.02	97.13	89.52	69.15
RC2-02	76.34	149.17	127.67	101.26	76.35	90.75	**49.32**

In order to support our hypothesis that "*if the selection mechanism affects the performance of Multi-GA, then a good selection mechanism can enhance its performance*", we perform a statistical test to identify whether the results of these algorithms are significantly different or not. Table 7 shows the ranking of the Multi-GA-RS, Multi-GA-BS, Multi-GA-SBCR, Multi-GA-CBSR, Multi-GA-RW, Multi-GA-SUS and Multi-GA-TS according to the Friedman test (the lower the value the higher the rank). The last two rows in Table 8 signify the p-value of Friedman and Iman-Davenport statistical tests. The tabulated results showed that, the Multi-GA-CBSR had the lowest value, so it is ranked the first, whilst, Multi-GA-BS, Multi-GA-SBCR, Multi-GA-RS, Multi-GA-RW, Multi-GA-SUS and Multi-GA-TS are in the 2nd, 3rd, 4th, 5th and 6th ranks, respectively.

Table 8 The ranking of the Multi-GAs

Algorithm	Ranking
Multi-GA-CBSR	**1.7**
Multi-GA-BS	2.25
Multi-GA- SBCR	2.58
Multi-GA-RS	4.92
Multi-GA-RW	5.17
Multi-GA-SUS	5
Multi-GA-TS	6.92
Friedman test(*p-value*)	0.00
Iman-Davenport(*p-value*)	0.00

The bold fonts shows the lowest value

Table 9 The adjusted p-value for the Multi-GA-BR versus other Multi-GAs algorithms

Multi-GA-CBSR *vs*	p-value
Multi-GA-BS	0.219303
Multi-GA-SBCR	0.108197
Multi-GA-RS	**0.000021**
Multi-GA-RW	**0.000006**
Multi-GA-SUS	**0.000014**
Multi-GA-TS	**0.00**

The bold fonts demonstrate that the results are statistically significant

A Wilcoxon test with critical level of 0.05 is conducted to verify whether the Multi-GA-CBSR is statistically better than the others or not. Table 9 shows that the Multi-GA-CBSR is statistically better than Multi-GA-RS, Multi-GA-RW, Multi-GA-SUS and Multi-GA-TS (adjusted p -value < 0.05), yet it is not statistically significant compared to Multi-GA-BS and Multi-GA-SBCR.

To summarize, the results revealed the efficiency of utilizing multi parent crossover with CBSR selection mechanism in comparison with crossovers of multi parents with other selection mechanisms. This is possibly ascribed to the employment of the best individuals as the first and the second parents in multi parent crossover which improved the exploitation process via using the information of the best individuals in mating process.

5. CONCLUSION

The influence of selection mechanism on the performance of the genetic algorithm with multi parent crossover was investigated in this paper. To investigate this influence, seven selection mechanisms were utilized in this work, which resulted in seven different types of genetic algorithm with multi parent crossover (Multi-GAs). The results obtained from the application of these algorithms were verified on Solomon's VRPTW benchmark. Results showed that Multi-GA-CBSR was more efficient in obtaining much better results than those obtained by others. This is possibly ascribed to the employment of the CBSR selection mechanism in selecting parents as it can select the best individuals from a population to represent the first and the second parents while the third parent was selected randomly. This is because choosing the information of the best parents in multi parent crossover can enhance the Multi-GA exploitation ability while the random selection of the third parent in addition to the mutation operator can enhance the exploration ability of Multi-GA. Consequently, the probability of attaining the desired balance between exploitation and exploration increased. To support these results, a statistical test was conducted. From a statistical stand point, the Multi-GA obtained better results when CBSR selection mechanism was applied.

REFERENCES

[1]O. Bräysy and M. Gendreau, "Vehicle routing problem with time windows, Part I: Route construction and local search algorithms," Transportation science, vol. 39, pp. 104-118, 2005a.

[2]G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," Management science, pp. 80-91, 1959.

[3]J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, and F. Soumis, "VRP with time windows," The vehicle routing problem, vol. 9, pp. 157-193, 2001.

[4]E. T. Yassen, M. Ayob, M. Z. A. Nazri, and N. R. Sabar, "Meta-harmony search algorithm for the vehicle routing problem with time windows," Information Sciences, vol. 325, pp. 140-158, 2015.

[5]Y. J. Gong, J. Zhang, O. Liu, R. Z. Huang, H. S. H. Chung, and Y. H. Shi, "Optimizing the Vehicle Routing Problem With Time Windows: A Discrete Particle Swarm Optimization Approach," IEEE Transactions on Systems Man and Cybernetics-Part C-Applications Reviews, vol. 42, p. 254, 2012b.

[6]E. T. Yassen, M. Ayob, M. Z. Ahmad Nazri, and N. R. Sabar, "A hybrid meta-heuristic algorithm for vehicle routing problem with time windows," International Journal on Artificial Intelligence Tools, vol. 24, p. 1550021, 2015.

[7]Y. Rochat and É. D. Taillard, "Probabilistic diversification and intensification in local search for vehicle routing," Journal of heuristics, vol. 1, pp. 147-167, 1995.

[8]R. A. Russell and W. C. Chiang, "Scatter search for the vehicle routing problem with time windows," European Journal of Operational Research, vol. 169, pp. 606-622, 2006.

[9]H. Nazif and L. Lee, "Optimized crossover genetic algorithm for vehicle routing problem with time windows," American journal of applied sciences, vol. 7, pp. 95-101, 2010.

[10] Y. Wang, "A Hybrid Approach Based on Ant Colony System for the VRPTW," 2012, pp. 327-333.

[11] J. H. Holland, Adaptation in natural and artificial systems: University of Michigan press, 1975.

[12] W. Gong, Á. Fialho, Z. Cai, and H. Li, "Adaptive strategy selection in differential evolution for numerical optimization: an empirical study," Information Sciences, vol. 181, pp. 5364-5386, 2011b.

[13] G. Zapfel, R. Braune, and M. Bogl, Metaheuristic search concepts: Springer, 2010.

[14] B. Kallehauge, J. Larsen, O. B. G. Madsen, and M. M. Solomon, "Vehicle routing problem with time windows," Column generation, pp. 67-98, 2005.

[15] K. Tan, T. Lee, K. Ou, and L. Lee, "A messy genetic algorithm for the vehicle routing problem with time window constraints," in

Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, 2001c, pp. 679-686.

[16] S. R. Thangiah, Vehicle routing with time windows using genetic algorithms: Citeseer, 1993.

[17] S. R. Thangiah, "An adaptive clustering method using a geometric shape for vehicle routing problems with time windows," in Proceedings of the 6th International Conference on Genetic Algorithms, 1995, pp. 536-545.

[18] E. G. Talbi, Metaheuristics from design to implementation: Wiley Online Library, 2009.

[19] I. Oliver, D. Smith, and J. Holland, "C, 1987. A Study of Permutation Crossover Operators on the Travelling Salesman Problem," in Genetic Algorithms and their Application:Proceedings of the 2nd International Conference on Genetic Algorithms.

[20] Y. Nagata, "Edge assembly crossover for the capacitated vehicle routing problem," Evolutionary Computation in Combinatorial Optimization, pp. 142-153, 2007.

[21] Z. Lü, J. K. Hao, and F. Glover, "A Study of Memetic Search with Multi-parent Combination for UBQP," Evolutionary Computation in Combinatorial Optimization, pp. 154-165, 2010.

[22] Y. Wang, Z. Lü, and J. K. Hao, "A study of multi-parent crossover operators in a memetic algorithm," Parallel Problem Solving from Nature–PPSN XI, pp. 556-565, 2011.

[23] E. T. Yassen, M. Ayob, M. Z. A. Nazri, and N. R. Sabar, "Multi-parent insertion crossover for vehicle routing problem with time windows," in Data Mining and Optimization (DMO), 2012 4th Conference on, 2012, pp. 103-108.

[24] J. Berger, M. Barkaoui, and O. Braysy, "A route-directed hybrid genetic approach for the vehicle routing problem with time windows," Infor-Information Systems and Operational Research, vol. 41, pp. 179-194, 2003.

[25] D. E. Goldberg, Genetic algorithms in search, optimization, and machine learning vol. 412: Addison-wesley Reading Menlo Park, 1989.

[26] K. C. Tan, L. L. Hay, and O. Ke, "Hybrid genetic algorithms in solving vehicle routing problems with time window constraints," Asia-Pacific Journal of Operational Research, vol. 18, pp. 121-131, 2001b.

[27] S. Jung and B. R. Moon, "A Hybrid Genetic Algorithm For The Vehicle Routing Problem With Time Windows," in GECCO, 2002, pp. 1309-1316.

[28] G. B. Alvarenga and G. R. Mateus, "Hierarchical tournament selection genetic algorithm for the vehicle routing problem with time windows," in Hybrid Intelligent Systems, 2004. HIS'04. Fourth International Conference on, 2004, pp. 410-415.

[29] B. Minocha and S. Tripathi, "Solution of Time Constrained Vehicle Routing Problems using Multi-Objective Hybrid Genetic Algorithm," Int. J. Comput. Sci. Inf. Technol., vol. 2, pp. 2671-2676, 2011.

[30] Y.-J. Shi, F.-W. Meng, and G.-J. Shen, "A modified artificial bee colony algorithm for vehicle routing problems with time windows," Information Technology Journal, vol. 11, p. 1490, 2012.

[31] T. Vidal, T. Gabriel Crainic, M. Gendreau, and C. Prins, "A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows," Computers & Operations Research, 2013.

[32] J.-Y. Potvin and S. Bengio, "The vehicle routing problem with time windows part II: genetic search," INFORMS Journal on computing, vol. 8, pp. 165-172, 1996.

[33] A. Le Bouthillier and T. G. Crainic, "A cooperative parallel meta-heuristic for the vehicle routing problem with time windows," Computers & Operations Research, vol. 32, pp. 1685-1708, 2005.

[34] H. Wee-Kit , J. C. Ang, and A. Lim, "A HYBRID SEARCH ALOGRITHM FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS," International Journal on Artificial Intelligence Tools, vol. 10, pp. 431-449, 2001.

[35] H. Gehring and J. Homberger, "A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows," in Proceedings of EUROGEN99, 1999, pp. 57-64.

[36] H. Gehring and J. Homberger, "Parallelization of a two-phase metaheuristic for routing problems with time windows," Journal of Heuristics, vol. 8, pp. 251-276, 2002.

[37] D. Mester, "An evolutionary strategies algorithm for large scale vehicle routing problem with capacitate and time windows restrictions," in Conf. Mathematical and Population Genetics, University of Haifa, Israel, 2002.

[38] G. Zapfel, R. Braune, M. Bogl, and E. Corporation, Metaheuristic Search Concepts: Springer, 2010.

[39] H. M. Pandey, A. Shukla, A. Chaudhary, and D. Mehrotra, "Evaluation of genetic algorithm's selection methods," in Information Systems Design and Intelligent Applications, ed: Springer, 2016, pp. 731-738.

[40] C.-K. Ting, C.-H. Su, and C.-N. Lee, "Multi-parent extension of partially mapped crossover for combinatorial optimization problems," Expert systems with applications, vol. 37, pp. 1879-1886, 2010.

[41] Y. W. Foo, C. Goh, H. C. Lim, and Y. Li, "Evolutionary Neural Network Modeling for Energy Prediction of Cloud Data Centers," in International Symposium on Grids and Clouds, 2015.

[42] A. J. Kulkarni, G. Krishnasamy, and A. Abraham, "Cohort Intelligence for Solving Travelling Salesman Problems," in Cohort Intelligence: A Socio-inspired Optimization Method, ed: Springer, 2017, pp. 75-86.

[43] B. B. Akay and D. Karaboga, "Artificial Bee Colony Algorithm Variants on Constrained Optimization," An International Journal of Optimization and Control: Theories & Applications (IJOCTA), vol. 7, pp. 98-111, 2017.

[44] J. E. Baker, "Reducing bias and inefficiency in the selection algorithm," in Proceedings of the second international conference on genetic algorithms, 1987, pp. 14-21.

[45] W. La Cava, S. Silva, L. Vanneschi, L. Spector, and J. Moore, "Genetic programming representations for multi-dimensional feature learning in biomedical classification," Spring, Amsterdam, e Netherlands. To Appear, 2017.

[46] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," Operations research, pp. 254-265, 1987.

[47] C. Yang, Z.-x. Guo, and L.-y. Liu, "Comparison Study on Algorithms for Vehicle Routing Problem with Time Windows," in Proceedings of the 21st International Conference on Industrial Engineering and Engineering Management 2014, 2015, pp. 257-260.

[48] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," Information Sciences, vol. 180, pp. 2044-2064, 2010.

[49] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "A Dynamic Multi-Armed Bandit-Gene Expression Programming Hyper-Heuristic for Combinatorial Optimization Problems," IEEE transactions, vol. 45, pp. 217 - 228, 2014.

أثر تقنيات الاختيار على كفاءة التهجين متعدد الآباء

عصام طه ياسين
etalheety@gmail.com

الخلاصة

أثبت التهجين متعدد الأباء قدرته على حل العديد من مشاكل التحسين الصعبة والمعقدة مثل مشكلة البائع المتجول (Traveling salesman problem) و مشكلة النقل (Vehicle routing problem). ان نجاح هذا النوع من التهجين يعود لقابليته على تعزيز كفاءة البحث من خلال تبادل معلومات عدة أباء (أكثر من أبوين) وتوريثها ألى الأبناء. تتم عملية أختيار الأباء باستخدام واحدة من تقنيات الأختيار (Selection mechanisms) . أن أختيار أكثر الأباء ملائمة في عملية التهجين من الممكن أن يقود الى تحسين كفاءة الخوارزمية الجينية (Genetic Algorithm) . لذلك هذا البحث يتناول أثر تقنية الأختيار على كفاءة التهجين متعدد الأباء من خلال أختبار سبع تقنيات أختيار مختلفة. تم فحص أداء الخوارزميات المقترحة بأستخدام (Solomon VRPTW Benchmark) . أظهرت نتائج الأختبار تفوق التهجين متعدد الأباء الذي يستخدم تقنية الأختيار التي تعتمد أفضل الأفراد ليكونوا أباء في عملية التهجين. هذا يوضح ان اعتماد افضل الاباء في عملية التهجين متعدد الأباء يؤدي الى تعزيز عملية البحث والحصول على حلول جيدة.