A hybrid chemical reaction optimisation algorithm for solving the DNA fragment assembly problem

1st Naima Saidi
MISC Laboratory
Constantine 2 University
Constantine, Algeria
naima.saidi@univ-constantine2.dz

2nd Abdesslem Layeb
MISC Laboratory
Constantine 2 University
Constantine, Algeria
layeb.univ@gmail.com

***Abstract*—The DNA Fragment Assembly Problem (FAP) is a combinational optimisation problem in bioinformatics which is the process of reconstructing the original DNA sequence from a set of fragments produced by a sequencing machine. It is an NP-Hard problem. Therefore, finding an exact solution in a polynomial-time is impossible. Metaheuristics-based algorithms can be used to provide a good solution in reasonable time. In this paper, we have applied a Chemical Reaction Optimisation (CRO) algorithm combined with Simulated Annealing (SA) to the DNA fragment assembly problem. The experimental results showed that CRO+SA is very competitive with the state-of-the-art algorithms for this problem.**

***Index Terms*—Bioinformatics, DNA Fragment Assembly Problem, Chemical Reaction Optimisation, Simulated Annealing**

I. Introduction

The deoxyribonucleic acid (DNA) is a double stranded helix that contains genetic information needed for the development and functioning of almost all cells in a living organism. Each strand is constructed from four types of nucleotides: Adenine, Cytosine, Guanine, and Thymine. To determine the sequence of these nucleotides, the process of DNA sequencing is applied. Since current DNA sequencing technologies are not able to read the whole DNA sequence, only much shorter fragments called "reads", the DNA fragment assembly is needed to reconstruct the original DNA sequence from these reads.

The process of DNA sequencing starts with duplicating the original DNA sequence, then each copy is cut into short fragments at random points. After that, this biological material is converted to sequences of Ts, Gs, Cs, and As using a sequencing machine; this process is referred to as the shot-gun sequencing. After the reads are obtained, an assembly approach is followed to merge these reads into a longer DNA sequence.

The main approaches to the DNA fragment assembly problem are: the Overlap-Layout-Consensus (OLC) which is especially used for assembling long reads obtained by the Sanger sequencing or the third generation sequencing, and the De Brujin graph [1], this approach became popular for assembling the short reads produced by the next generation sequencing.

Unfortunately, the DNA fragment assembly is an NP-Hard problem [2], even with the elimination of sequencing errors and the difficulties caused by the repetitive structure of genomes. Therefore, metaheuristic approaches are employed to find good solutions efficiently. Chemical reaction optimization (CRO) is a powerful population-based optimization algorithm proposed by [3]. It mimics what happens to molecules in a chemical reaction system microscopically. The CRO is a discrete metaheuristic which made it suitable for the DNA fragment assembly problem. It has been successfully applied for several combinatorial and real world optimization problems such as : task scheduling in grid computing [4], the 01 knapsack problem [5], max flow problem [6], the vehicle routing problem [7], the energy conserving of sensor nodes in the design of wireless sensor networks [8], clustering algorithms for wireless sensor networks [9], and multiple sequence alignment [10].

In this paper, a chemical reaction optimization algorithm combined with a simulated annealing-based local search has been proposed to solve the DNA FAP. The simulated annealing-based local search have been used to enhance the final solution obtained by the CRO algorithm. We have validated our algorithm by using three set of benchmarks: Genfrag, Dnagen, and the f-seires. The experimental results show that the algorithm can get better overlap score than other metaheuristics-based approach.

The remainder of this paper is organized as follows. In section II, we give some basic concepts about the DNA fragment assembly problem. Section III presents the CRO algorithm. The manner of applying the CRO+SA on the DNA fragment assembly problem is detailed in section IV. Section V presents and discusses the experimental results obtained from applying the proposed approach on three set of benchmarks. Finally, Section VI concludes the paper.

II. The DNA fragments assembly problem

The DNA fragments assembly is one of the most difficult phases of any DNA sequencing project. Due to the fact that long DNA sequence cannot be accurately and rapidly sequenced. DNA sequencing provides the necessary information about the overlap to combine the reads back together. Therefore, the ultimate goal is to obtain a sequence as close as possible to the original one [11].

The OLC approach for the DNA fragment assembly problem proceeds in three phases:

1) *Overlap:* It consists in finding the longest common overlapping between the suffix of a sequence and the prefix of another one. This task requires the comparison of all possible pairs of fragments. It is usually tackled with a dynamic programming algorithm applied to semi-global alignments such as Smith-Waterman algorithm [12].
2) *Layout:* The goal of this step is ordering the fragments to maximise the overlap scores calculated in the previous phase. This NP-Hard problem [2] is the most difficult phase of the OLC approach.
3) *Consensus:* To determine the complete DNA sequence using the layout generated in the layout phase. The consensus is usually generated by applying the majority rule. For measuring the quality of a consensus, we can use Eq. 1, where n is the number of fragments in the target sequence. The coverage measures the data redundancy.

$$Coverage = \frac{\sum_{i=1}^{n} \text{length of the fragment i}}{\text{target sequence length}} \quad (1)$$

Since the DNA fragments assembly problem is an NP-Hard problem, Most assemblers are based on variations of a greedy algorithms, such as Phrap [13], TIGR Assembler [14], Celera Assembler [15], Velvet [16], and ABySS [17]. However, most proposed metaheuristics dealt with the layout phase of the OLC approach. Such algorithms based on Simulated annealing [18], tabu search [19], in [20], the authors have proposed a problem aware local search algorithm (PALS) to solve the problem with the object of achieving a maximum overlap value and a minimum number of contigs, which is a sequence of fragments with an overlap greater than a threshold between them. In [21],the authors have proposed two modifications to PALS to improve its accuracy and efficiency. In [22], the authors have studied the performance of different genetic algorithm operators for the DNA fragment assembly problem. They found that the edge-recombination crossover used with conjunction with two specialised operators-which manipulate the contigs rather than the individual fragments, perform best. Luque and Alba in [18] have proposed a Genetic Algorithm, Scatter Search and CHC algorithm for solving the DNA FAP. Due to its difficulty, scientists have opted to hybrid methods to solve this problem. Different hybridisation with PALS were proposed in the literature: GA [23], cellular GA [24], simulated Annealing [25]. The authors in [26] have proposed a hybrid PSO algorithm (HPSO). They used a a tabu search algorithm for initialising the particels and a simulated annealing algorithm to improve the best solution obtained by the PSO algorithm. The authors in [27] have presented an Artificial Bee Colony (ABC) algorithm and Queen-bee Evaluation based on Genetic Algorithm (QEGA) to tackle the problem of DNA fragment assembly for noisy and noiseless instances. In [28] the authors have proposed two algorithms namely genetic algorithm with simulated annealing (GA+SA) and genetic algorithm with hill climbing (GA+HC). The authors in [29] have presented a hybrid meta-heuristic based on Simulated Annealing and a genetic crossover operator. In [30] the authors have proposed a memetic PSO algorithms based on two initialization methods: the Tabu Search (TS) and simulated annealing (SA). In [31] the authors have proposed a collection of GA variations (Recentering-Restarting, Ring Species, and Island Model) in combination with each other. These algorithms also used heuristics, namely, 2-Opt and the Lin-Kernighan Heuristic. They studied the performance of these algorithms with different solution representations. The most recent metaheuristic used for solving the DNA FAP is the crow search algorithm in hybridisation with the improved PALS (PALS2Many*) in [32]. The authors have used a modified version of the ordered crossover operator to adapt the continuous crow search algorithm to the DNA FAP; and in the local search they used only the movement that increase the overlap values not the number of contigs. In [11] a parallel hybrid algorithm between Particle Swarm Optimization (PSO) and Differential Evolution (DE), (PPSO+DE) have been proposed.

III. The chemical reaction optimisation algorithm

CRO is a recent population-based nature inspired metaheuristic. It mimics the process of transforming a set of unstable molecules through a sequence of elementary reactions into stable products. Every molecule has a molecular structure (w), a potential energy (PE), and a kinetic energy (ke); and optional attributes that depends on the problem. Such as: the number of hits (NumHit), the minimum PE (MinPE), and the minimum hit number (MinHit). Illustrations of these attributes are presented in Table I . There are four types of reactions in the CRO, grouped into uni-molecular and multi-molecular:

1) Uni-molecular reactions
   - On-wall ineffective collision
   - Decomposition
2) Inter-molecular reactions
   - Inter-molecular ineffective collision
   - Synthesis

The four elementary reactions are described as follows.

On-wall ineffective collision: In this reaction, a molecule w hits the wall of the container, then bounces away remaining in one single unit. As a result, a new molecule w' in the neighbourhood of the first one is generated. The change is allowed only if

$$PE_w + KE_w \geq PE_{w'} \quad (2)$$

we get

$$KE_{w'} = (PE_w + KE_w - PE_{w'}) * q \quad (3)$$

where $q \in [KELossRate, 1]$, and $(1 - q)$ represents the fraction of KE lost to the environment when it hits the wall.

Decomposition: In the decomposition, a molecule hits the wall and then decomposes into two or more pieces. in this paper, we assume that a molecule w produces two molecules

TABLE I
THE ATTRIBUTES OF A MOLECULE USED IN CRO WITH ITS MEANING

Attribute	Meaning
Molecular structure (w)	Represents a solution of the problem.
Potential energy (PE)	Defines the objective function value of the corresponding solution represented by w.
Kinetic energy (KE)	A non-negative number, it quantifies the tolerance of the system accepting a worse solution than the existing one.
Number of hits ($numHit$)	Indicates the total number of collisions the molecule has been involved in.
Minimum PE ($minPE$)	Means the best objective function value that the molecule has experienced.
Minimum hit number ($minHit$)	Records the number of hits when a molecule obtains its latest minimum PE. It is an abstract notation of the period of time a molecule has stayed in a stable state.

w_1 and w_2. To perform the decomposition, the criterion $(NumHit - MinHit > \alpha)$ has to be fulfilled. The change is allowed if

$$PE_w + KE_w \geq PE_{w1} + PE_{w2} \tag{4}$$

or

$$PE_w + KE_w + buffer \geq PE_{w1} + PE_{w2} \tag{5}$$

if 4 is satisfied we get

$$KE_{w1} = temp1 \times q$$

and

$$KE_{w2} = temp1 \times (1 - q)$$

where $temp1 = PE_w + KE_w PE_{w1} - PE_{w2}$ and q is randomly generated from the interval $[0, 1]$. And if 5 is satisfied we get

$$KE_{w1} = (temp1 + buffer) \times q_1 \times q_2$$

and

$$KE_{w2} = (temp1 + buffer - KE_{w1}) \times q_3 \times q_4$$

Then the buffer is updated by

$$buffer = buffer + temp1 + KE_{w1} + KE_{w2}$$

where q_1, q_2, q_3, and q_4 are randomly generated from the interval $[0, 1]$. If 4 and 5 are not satisfied, the decomposition reaction does not hold and the molecule has its original structure w.

Inter-molecular ineffective collision: In this reaction, two or more molecules (assume two) collide with each other and bounce away. This reaction is similar to the On-wall ineffective collision, we generate two molecule w_1' and w_2' in the neighbourhoods of w_1 and w_2 respectively. The change is allowed if

$$PE_{w1} + PE_{w2} + KE_{w1} + KE_{w2} \geq PE'_{w1} + PE'_{w2}. \tag{6}$$

Let $temp2 = (PE_{w_1} + PE_{w_2} + KE_{w_1} + KE_{w_2})(PE'_{w_1} + PE'_{w_2}$.
We get

$$KE'_{w_1} = temp2 \times q$$

and

$$KE'_{w_2} = temp2 \times (1 - q)$$

where p is a random number uniformly generated from the interval $[0, 1]$.

Synthesis: A synthesis happens when multiple (assume two) molecules hit against each other and fuse together. We generate a new molecule w' a quite different from the two original molecules w_1 and w_2. The condition for synthesis is $(KE_{w_1} < \beta \quad and \quad KE_{w_2} < \beta)$. The change is allowed only if

$$PE_{w_1} + PE_{w_2} + KE_{w_1} + KE_{w_2} \geq PE_{w'}. \tag{7}$$

We get

$$KE_{w'} = PE_{w_1} + PE_{w_2} + KE_{w_1} + KE_{w_2} - PE_{w'}.$$

IV. CRO+SA FOR THE DNA FRAGMENT ASSEMBLY PROBLEM

As previously mentioned, the present work uses a chemical reaction optimisation algorithm combined with a local search for solving the layout phase in the OLC approach for the DNA fragment assembly problem. In fact, the use of exact methods for finding the optimal layout would be very time-consuming, which motivates the use of metaheuristics. In this section, we start by defining the aspects of the solution presentation and the objective function, then we present the use of the CRO and the simulated annealing algorithm to tackle the DNA fragment assembly problem.

A. Solution representation

A solution in CRO is presented by the molecular structure of a molecule. Solutions are represented by a permutation of N (the number of reads) integer encoding a sequence of fragment identifiers.

B. Objective function

Chemical reaction optimisation was originally designed to tackle minimisation problems. In this paper, we adopt the fitness function proposed in [22] by adding a minus sign to it, as shown in Eq. 8

$$Minimise : F(S) = -\sum_{j=1}^{n-1} w(f_j, f_{j+1}) \tag{8}$$

Where S denotes a molecule, and $w(f_j, f_{j+1})$ is the overlap score between the two adjacent fragments, calculated using a dynamic programming semi-global alignment algorithm. The settings used were 1 for a match, 3 for a mismatch, and 2 for a gap [33].

C. The search process

In our approach, a chemical reaction optimisation algorithm is used to search for the best layout that minimises the objective function Eq. 8. As summarised in the flowchart of Fig. 1, the algorithm starts by defining the initial population, and assigning values to the control parameters. After that, a number of iterations are performed. In each iteration, we perform one of the four collisions of the CRO, then we check for any new minimum fitness value and record it. After a certain number of iterations without improvements, we substitute the worst 20% of the population with new solutions generated by the same method as the initial population. This process helps to maintain the diversity of the population. After a maximum number of iterations performed without improvements, we finish the CRO process and use the simulated annealing to improve the best solution obtained. More details are presented in the following:

1) Population initialisation: To provide good and diverse solutions, we have combined two strategies to set the values of each solution. We used a greedy-based approach to set the values of 60% of a solution and we assigned the rest randomly from the fragments that had not been selected before. This approach serves a twofold purpose. Firstly, the greedy approach helps to speed up the search. Secondly, the random approach prevents the algorithm from stacking at a local minimum.

2) Elementary reactions: To choose one of the four elementary reactions, we first decide whether it is a uni-molecular or an inter-molecular collision. To do this, we generate a random number t, in the interval of [0, 1]. If t is larger than $molecoll$, it will result in an event of uni-molecular collision; Otherwise, an inter-molecular collision will take place. Next, we examine the criteria of decomposition or synthesis to decide which type of collision will take place.

1) *On-wall ineffective collision:* Since this reaction is used to make a small change in the molecular structure, it is done by selecting a fragment that does not have an overlap greater than a threshold with its adjacent fragments, or selecting a random one.To do this, we generate a random number t, in the interval of [0, 1]. If t is larger than 0.2, it will result in a random selection. Otherwise, we select an isolated fragment. Then, we search for the best position for it in the permutation. Figure 2 shows a graphical representation of on-wall ineffective collision.
2) *Decomposition:* In this reaction, we have used the half-total change operator [34]. we have produced two new solutions from an existing one by keeping one half of the existing solution for the new one and assigning the remaining half by piking randomly the values form the other half. Figure 3 depicts the decomposition collision.
3) *Inter-molecular ineffective collision:* In this operator, we generate two molecule w_1' and w_2' in the neighbourhoods of w_1 and w_2 respectively. to do this, we have used the two points crossover. we start by selecting randomly two points. Then, we swap all the fragments between the two points between the solutions w_1 and w_2 as shown in Figure 4.
4) *Synthesis:* We have used an enhanced edge recombination operator [35]. This operator emphasizes the adjacency information instead of the order or position of the fragments in the layout. To create a new solution, we use the information contained in the " edge table", which is an adjacency table listing the connections between the fragments found in the two molecules. Figure 5 shows a graphical representation of the synthesis collision.

3) The simulated annealing: The Simulated Annealing (SA) algorithm is a well-known metaheuristic for solving combinatorial optimisation problem, proposed in [36]. SA is a local search algorithm inspired from the cooling process of molten metals through annealing to find the optimal solution. It has been successfully applied to many difficult optimization problems such as the hybrid vehicle routing problem [37] and the competitive single and multiple allocation hub location problems [38]. In our proposal, the SA starts with the solution provided by the CRO algorithm. It remains for some time at the same temperature while a fixed number of iterations is performed. Then, the temperature is cold down. In each iteration, we choose one of three operators:

1) Inversion: In this operator, we randomly select two points. Then, we reverse the order of the fragments between them as shown in Figure 6.
2) Specific inversion: we invert the orientation of one contig. To do this, we randomly select one point in the permutation and determine the contig that contains this fragment. Then, we reverse the order of the fragments in the selected contig. [22]. Figure 7 show an example of the specific inversion operator.
3) Transposition: this operator moves a contig to a position between two adjacent contigs [22]. we randomly select two points in different contigs to determine the contig that will be moved and its new position. Then, we move the contig as shown in figure 8.

V. Experimental results and discussion

In this section, we evaluate the performance of our algorithm, implemented in Matlab R2015a and tested on a personal computer with Intel(R) Core()TM i3-4005U CPU at 1.70GHz 1.70GHz, 4GB RAM, running on Windows 8.1 64-bit.

We have tested our algorithm through applying it on thirty benchmarks divided into three collections: GenFrag, DNAgen and f-series, Table II represents a summary of them. The first column represents the name of the instance, the second column contains the original DNA sequence of each benchmark, the third column is the length of the original DNA sequence, the rest of the columns are the coverage, the mean fragment length and the number of fragments. More details on these benchmarks can be found in [33]. All the benchmarks are available for downloading at http://chac.sis.uia.mx/fragbench/descargas.php. Table III shows the parameters settings of

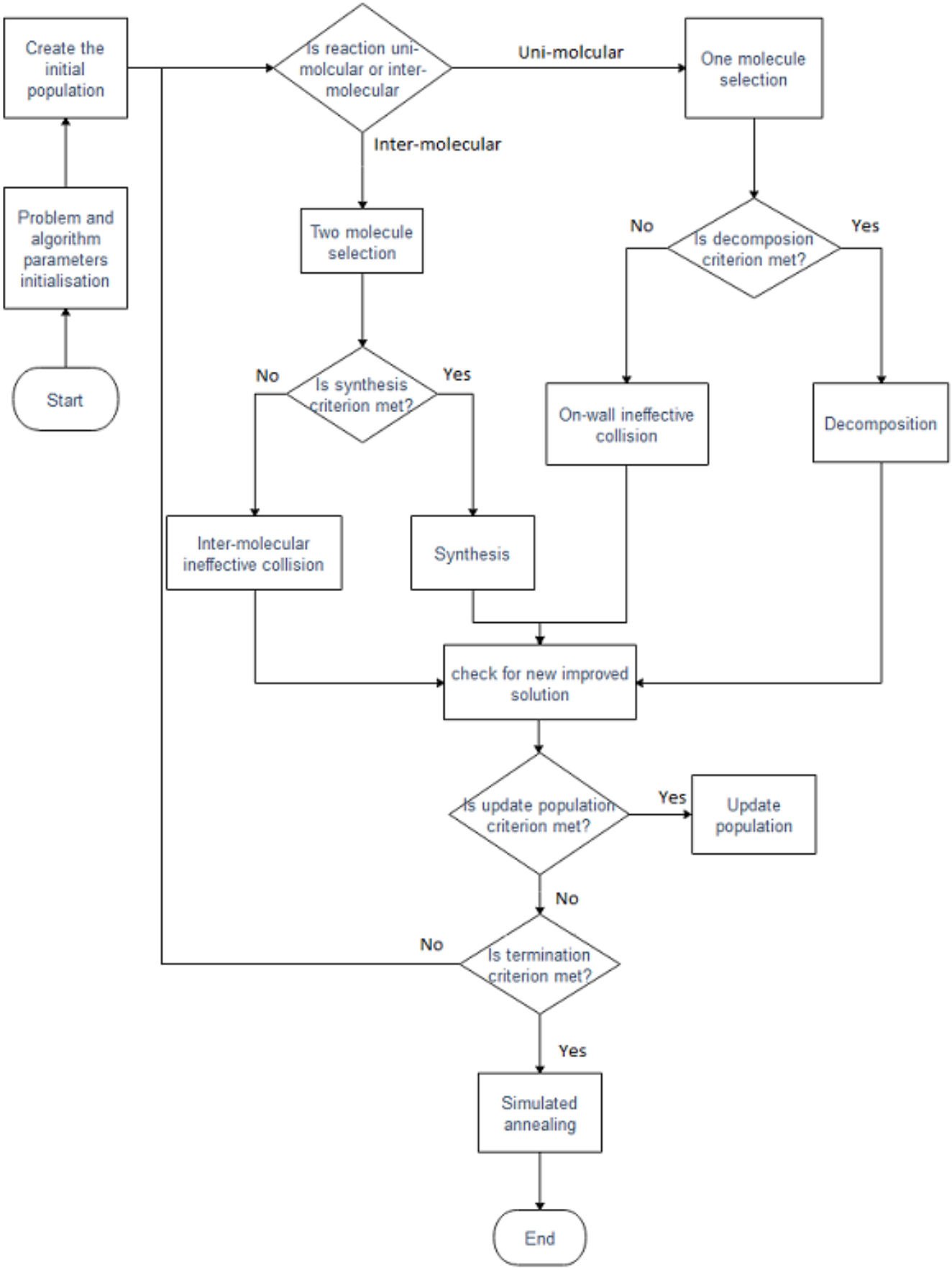

Fig. 1. The proposed CRO+SA algorithm for the DNA fragment assembly problem

TABLE II
DESCRIPTION OF THE THREE SET BENCHMARKS FOR THE DNA FAP USED IN THIS STUDY

Instances	Original DNA Seq.	original Seq. length	Coverage	Mean Frag. length	nbr of Frag.
GenFrag instances					
x60189(4)	A cluster of fibronectin type III repeats found in the human major histocompatibility complex class III region	3835	4	395	39
X60189(5)			5	286	48
X60189(6)			6	343	66
X60189(7)			7	387	68
M15421(5)	A human apolipoprotein B gene	10089	5	398	127
M15421(6)			6	350	173
M15421(7)			7	383	177
j02459(7)	Complete nucleotide sequence of the cohesive ends of bacteriophage lambda DNA	20000	7	405	352
bx842596(4)	Neurospora crassa DNA linkage group II BAC clone B10K17	77292	4	708	442
bx842596(7)			7	703	773
DNAgen instances					
acin1	a human microbion bacterium ATCC 49176	2170	26	182	307
acin2		147200	3	1002	451
acin3		200741	3	1001	601
acin5		329958	2	1003	751
acin7		426840	2	1003	901
acin9		156305	7	1003	1049
f-series					
f25-305	-	-	-	305	25
f25-400	-	-	-	400	25
f25-500	-	-	-	500	27
f50-315	-	-	-	315	50
f50-412	-	-	-	412	50
f50-498	-	-	-	498	50
f100-307	-	-	-	307	100
f100-415	-	-	-	415	100
f100-512	-	-	-	512	100
f508-354	-	-	-	354	508
f635-350	-	-	-	350	635
f737-355	-	-	-	355	737
f1343-354	-	-	-	354	1343
f1577-354	-	-	-	354	1577

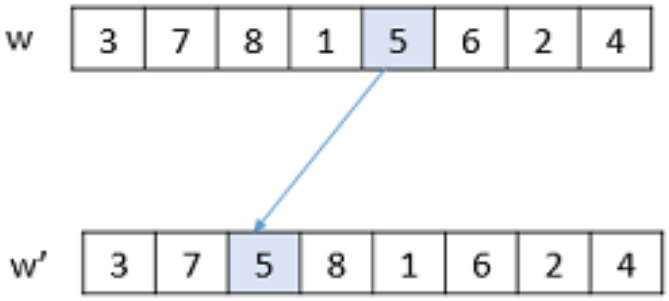

Fig. 2. Illustration of on-wall ineffective collision

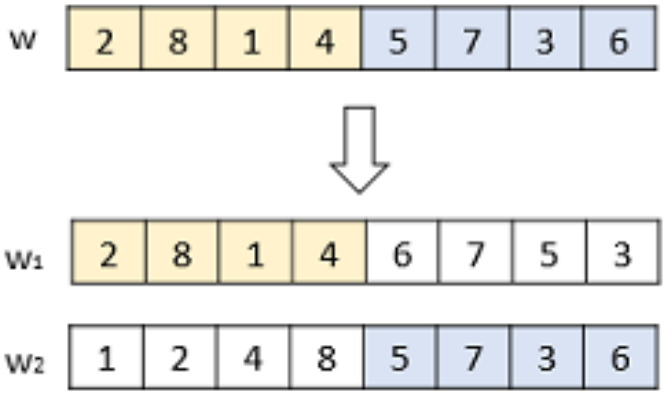

Fig. 3. Illustration of decomposition collision

the CRO and SA algorithms. The parameters values were determined empirically by observing the results obtained for different settings.

Table IV gives the results of the first version of the CRO and the hybrid CRO+SA. It shows the results of the two algorithms in terms of the best, average, worst and standard deviation values obtained over 30 independent runs. The optimal column presents the optimal fitness values obtained by the LKH algorithm [39], the LKH results were presented in [33]. The solutions that reached the optimal values are presented in bold. The purpose of this experiment is to show the enhancement of the CRO algorithm by a local search such as the simulated annealing. As we can see from Table IV, the simulated annealing algorithm had clearly improved the results obtained by the CRO. These results validate that the use of a

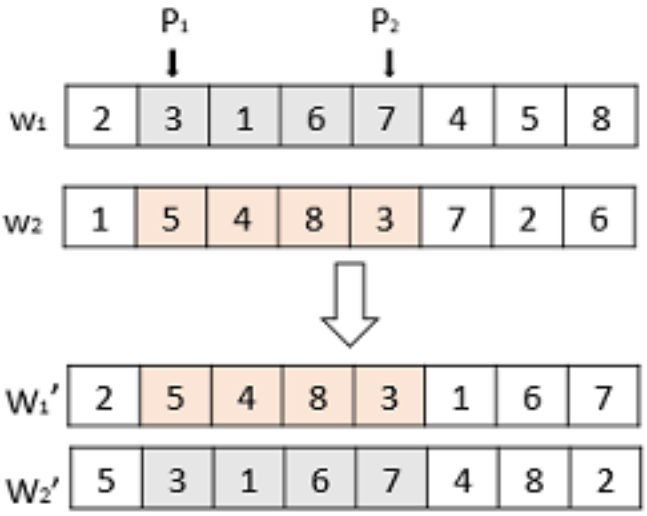

Fig. 4. Illustration of inter-molecular ineffective collision

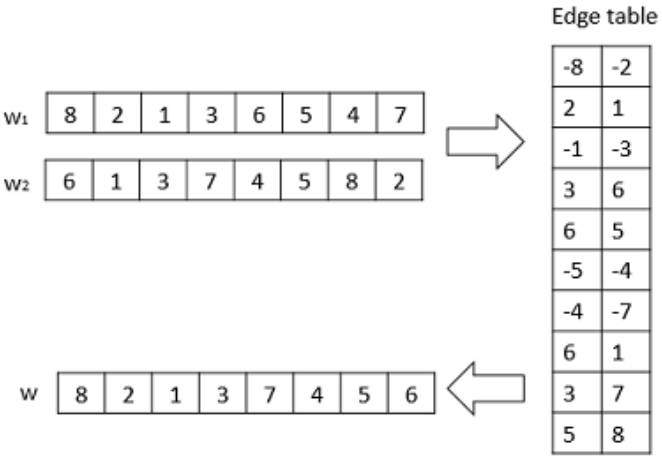

Fig. 5. Illustration of synthesis collision

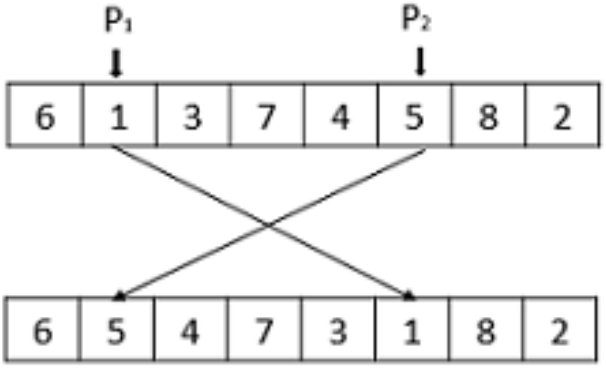

Fig. 6. Illustration of inversion operator

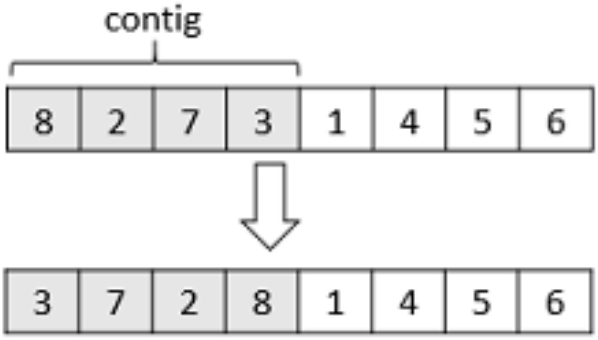

Fig. 7. Illustration of specific inversion operator

.

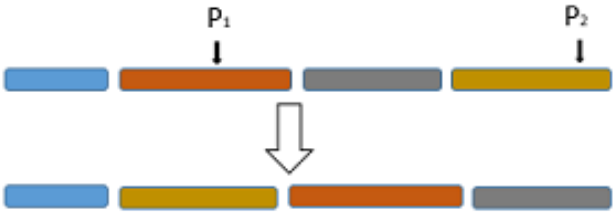

Fig. 8. Illustration of transposition operator

local search with large-scale neighbourhood operators, which manipulate the contgs [22], improves the results. The low rates at which the specific operators are applied is due to the fact that the number of contigs in the solutions obtained by the CRO are low as shown in Table V. Furthermore, the random inversion operator also allows a large scale modification in a solution and prevent the generation of redundant solutions contrary to the specific operators.

As mentioned above, Table V gives a summary of the number of contigs obtained by each algorithm. The threshold used to calculate the number of contigs is thirty. Simulated

TABLE III
THE PARAMETERS OF THE CRO+SA ALGORITHM

Parameter	Value
CRO parameters	
size of the initial population	50
initial kinetic energy	1000
α	400
β	10
kelossrate	0.2
molecoll	0.25
SA parameters	
Initial temperature	50
Inversion rate	0.85
Specific inversion rate	0.06
Transposition rate	0.09

TABLE IV
THE RESULTS OF THE CRO AND CRO+SA ALGORITHMS

Benchmark	Optimal	CRO Best	Average	Worst	STD	CRO+SA Best	Average	Worst	STD
GenFrag									
x60189(4)	11478	**11478**	11275.733	10937	188.388	**11478**	**11478**	**11478**	0
X60189(5)	14161	13753	13603.2	13402	90.603	**14161**	**14161**	**14161**	0
X60189(6)	18301	18111	17684.966	17036	267.729	**18301**	**18301**	**18301**	0
X60189(7)	21271	20700	20194.033	19432	311.661	**21271**	21260.4	21210	19.796
M15421(5)	38746	37975	36395.4	34975	615.427	**38746**	38740	38706	12.165
M15421(6)	48052	46696	44406.3	42373	1103.849	**48052**	**48052**	**48052**	0
M15421(7)	55171	54019	51316.533	48276	1687.53	**55171**	55158.233	55130	11.732
j02459(7)	116700	111951	107675.466	97086	3793.992	**116700**	116677.8	116612	23.649
bx842596(4)	227920	217308	211594.2	191079	7873.797	227914	227682.7	227127	179.206
bx842596(7)	445422	421482	391629.766	373515	15175.729	444518	442822.533	441238	760.515
DNAgen									
acin1	47618	43127	41679.466	40475	677.354	47613	47496.1	47320	68.017
acin2	151553	146360	143683.233	141490	1061.346	151456	151394.366	151361	22.48
acin3	167877	160182	159036.866	157584	708.128	167228	166980.166	166793	132.64
acin5	163906	157814	156551.3	155153	613.182	163426	163326.4	163247	43.914
acin7	180966	175869	175173.7	174057	08.463	180220	180065.466	179947	62.272
acin9	344107	327919	325452.7	323377	1177.81	343385	343247.5	343093	76.949
f-series									
f25(305)	596	592	586.333	584	2.07	**596**	**596**	**596**	0
f25 (400)	777	774	766.033	761	2.938	**777**	**777**	**777**	0
f25(500)	921	916	911.2	905	2.773	**921**	**921**	**921**	0
f50(315)	1581	1540	1514.933	1493	10.639	**1581**	**1581**	**1581**	0
f50(412)	1573	1529	1507.233	1486	10.167	**1573**	1572.2	1570	0.979
f50(498)	1570	1540	1515.333	1498	11.542	**1570**	1569.466	1568	0.845
f100(307)	2793	2652	2623.5	2589	14.046	**2793**	2788.433	2785	1.994
f100(415)	2860	2714	2690.5	2659	14.216	**2860**	2854.933	2851	2.112
f100(512)	2732	2577	2541.766	2508	15.658	2731	2727.6	2723	2.374
f508(354)	18112	16665	16428.166	15979	130.376	18007	17986.333	17948	15.08
f635(350)	22498	20578	20076.966	18880	387.379	22358	22282.166	22240	25.486
f737(355)	25218	23056	22541.9	21766	281.075	24948	24899.566	24865	22.819
f1343(354)	49042	44313	42530.4	40997	849.78	48121	47994.866	47913	51.031
f1577(354)	57373	52131	50165.966	47770	1228.46	56024	55896.7	55731	82.088

annealing had significantly reduced the number of contigs for the GenFrag and DNAgen benchmarks. However, for the f-series benchmarks, the CRO and CRO+SA algorithms have given the same results for the three first small instances. For the next six instances, the two algorithms reached the same best results but the average number of contigs is improved after applying the SA algorithm. For the large instances in this collection, the SA has reduced the number of contigs in the solutions obtained by the CRO algorithm.

Table VI compares CRO+SA with the state-of-the-art algorithms for the DNA fragment assembly problem that have been applied on the same benchmarks. The algorithms presented in the table are LKH [39], PPSO+DE [11], QEGA [27], SA [40], PALS [20], SAX [29], RRG [31] and CSA+P2M [32]. The optimal solutions are presented in bold. We can see that the CRO+SA has given competitive results with LKH and CSA+P2M algorithms and has outperformed the rest of the algorithms in most instances.

Indeed, The statistical Friedman test of Figure 9 represents a comparison of the results of the algorithms presented in table VI for the GenFrag and DNAgen benchmarks. As it is shown in the Friedman test, there is not a significant difference between the CRO+SA, RRGA, CSA+P2M and the optimal solutions. Furthermore, the results of the hybrid genetic algo-

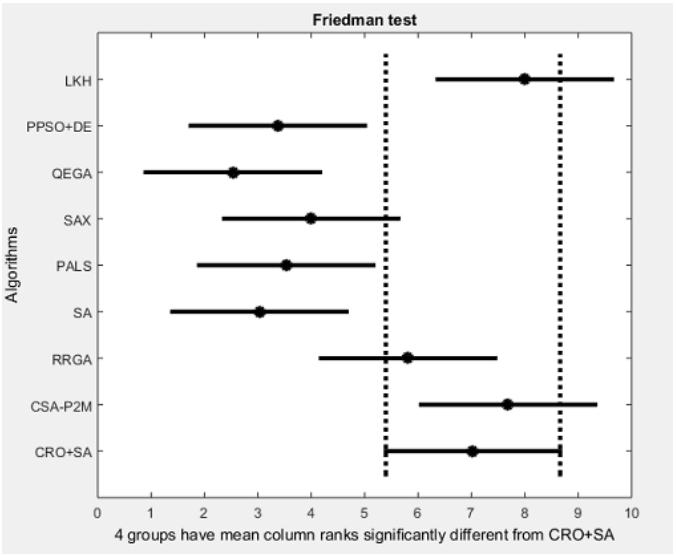

Fig. 9. Friedman test compares the algorithms presented in table VI for the GenFrag and DNAgen benchmarks

TABLE V
THE NUMBER OF CONTIGS OBTAINED BY THE CRO AND CRO+SA ALGORITHMS

Benchmark	CRO			CRO+SA		
	Best	Average	Worst	Best	Average	Worst
GenFrag						
x60189(4)	1	1.8	3	1	1	1
X60189(5)	2	3.333	5	2	2	2
X60189(6)	2	3	5	1	1	1
X60189(7)	2	3.366	6	1	1.166	2
M15421(5)	9	14.666	19	3	3.2	4
M15421(6)	7	13.166	20	2	2	2
M15421(7)	5	14	27	3	3	3
j02459(7)	16	28.566	57	2	2	2
bx842596(4)	33	48.2	71	9	10.333	14
bx842596(7)	37	84.233	112	5	9.833	14
DNAgen						
acin1	28	48.2	61	6	8.766	13
acin2	118	172.166	189	65	71.1	78
acin3	182	235.066	275	83	92.866	104
acin5	204	254.233	325	107	117.333	131
acin7	245	286.366	339	116	130.733	144
acin9	267	370.866	408	77	89.366	105
f-series						
f25(305)	19	19	19	19	19	19
f25 (400)	14	14	14	14	14	14
f25(500)	14	14	14	14	14	14
f50(315)	26	26.466	28	26	26	26
f50(412)	26	26.466	28	26	26	26
f50(498)	28	28.066	29	28	28	28
f100(307)	69	69.5	71	69	69	69
f100(415)	65	66.1	69	65	65	65
f100(512)	69	69.933	71	69	69	69
f508(354)	273	279.433	285	260	260.7	263
f635(350)	357	366	390	332	334.166	337
f737(355)	428	437.4	455	404	406.233	410
f1343(354)	707	749.033	781	657	662.933	672
f1577(354)	856	889.233	936	810	816.966	827

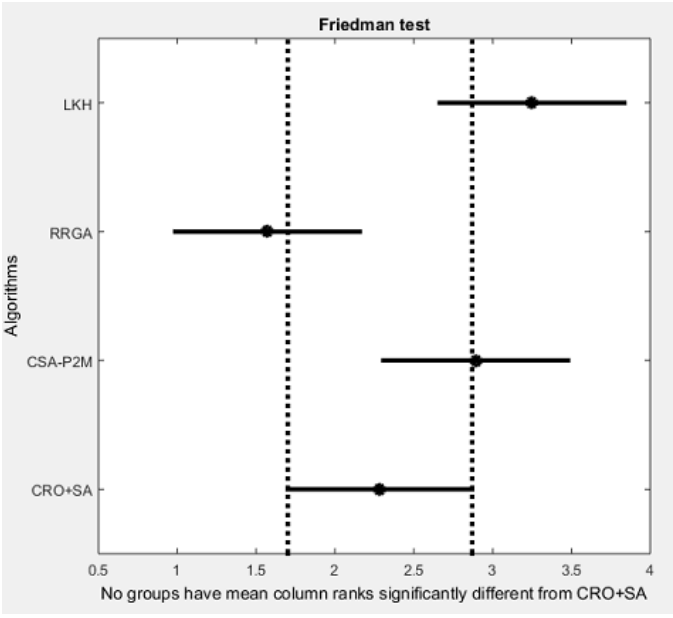

Fig. 10. Friedman test compares the LKH, RRGA, CSA+P2M and CRO+SA algorithms for the f-series benchmarks

rithm with the SA are close to the CRO+SA results. Figure 10 represents a statistical Friedman test compares the available results of the f-series benchmarks. The Friedman test shows that CRO+SA and CSA+P2M algorithms have no significant difference from the best known results.

The experimental results shows that the hybrid CRO algorithm with the simulated annealing gives good results for the DNA fragment assembly problem. As shown by the Friedman test, our algorithm ranks third for the three sets of benchmarks.

VI. CONCLUSION

In this work, we have proposed a hybrid CRO algorithm with simulated annealing for solving the layout phase of the OLC approach for the DNA fragment assembly problem. The experimental results show that combining CRO with SA can achieve the best overlap scores for sixteen benchmarks out of thirty benchmark data sets and very encouraging results for the rest of them.

As future work, it may be interesting to fine tuning and optimising our approach to improve the results. Since parallel implementation of the CRO can be done easily [34], a parallel version of the algorithm seems to be a possible option. It is also necessary to develop efficient optimisation algorithms to deal with noisy instances and real world scenarios.

REFERENCES

[1] Pavel A Pevzner, Haixu Tang, and Michael S Waterman. An eulerian path approach to dna fragment assembly. *Proceedings of the National Academy of Sciences*, 98(17):9748–9753, 2001.

[2] Pavel Pevzner. *Computational molecular biology: an algorithmic approach*. MIT press, 2000.

[3] Albert YS Lam and Victor OK Li. Chemical-reaction-inspired metaheuristic for optimization. *IEEE Transactions on Evolutionary Computation*, 14(3):381–399, 2010.

[4] Jin Xu, Albert YS Lam, and Victor OK Li. Chemical reaction optimization for task scheduling in grid computing. *IEEE Transactions on Parallel and Distributed Systems*, 22(10):1624–1631, 2011.

[5] Tung Khac Truong, Kenli Li, and Yuming Xu. Chemical reaction optimization with greedy strategy for the 0–1 knapsack problem. *Applied Soft Computing*, 13(4):1774–1780, 2013.

[6] Reham Barham, Ahmad Sharieh, and Azzam Sliet. Chemical reaction optimization for max flow problem. *IJACSA) International Journal of Advanced Computer Science and Applications*, 7(8), 2016.

[7] Min Li, Zhongxiang Liao, Yiming He, Jianxin Wang, Junwei Luo, and Yi Pan. Isea: iterative seed-extension algorithm for de novo assembly using paired-end information and insert size distribution. *IEEE/ACM transactions on computational biology and bioinformatics*, 14(4):916–925, 2017.

[8] PC Srinivasa Rao and Haider Banka. Novel chemical reaction optimization based unequal clustering and routing algorithms for wireless sensor networks. *Wireless Networks*, 23(3):759–778, 2017.

[9] PC Srinivasa Rao and Haider Banka. Energy efficient clustering algorithms for wireless sensor networks: novel chemical reaction optimization approach. *Wireless Networks*, 23(2):433–452, 2017.

[10] Rohit Kumar Yadav and Haider Banka. An improved chemical reaction-based approach for multiple sequence alignment. *CURRENT SCIENCE*, 112(3):527, 2017.

[11] Guillermo M Mallén-Fullerton and Guillermo Fernandez-Anaya. Dna fragment assembly using optimization. In *Evolutionary computation (CEC), 2013 IEEE congress on*, pages 1570–1577. IEEE, 2013.

[12] T Smith and M Waterman. ªidentification of common molecular subsequences. º j. *Molecular Biology*, 147:195–197, 1981.

[13] P Green. Phrap, version 1.090518. *Available at h ttp://phrap. org*, 2009.

[14] Granger G Sutton, Owen White, Mark D Adams, and Anthony R Kerlavage. Tigr assembler: A new tool for assembling large shotgun sequencing projects. *Genome Science and Technology*, 1(1):9–19, 1995.

TABLE VI
A COMPARISON OF THE CRO+SA AND EIGHT STATE-OF-THE-ART ALGORITHMS FOR SOLVING THE DNA FRAGMENT ASSEMBLY PROBLEM.

benchmark	LKH	PPSO+DE	QEGA	SA	PALS	SAX	RRGA	CSA+P2M	CRO+SA
GenFrag									
x60189(4)	**11478**	**11478**	11476	**11478**	**11478**	**11478**	**11478**	**11478**	**11478**
X60189(5)	**14161**	13642	14027	14027	14027	14027	**14161**	**14161**	**14161**
X60189(6)	**18301**	**18301**	18266	**18301**	**18301**	**18301**	**18301**	**18301**	**18301**
X60189(7)	**21271**	20921	21208	**21271**	21210	21268	21228	**21271**	**21271**
M15421(5)	**38746**	38686	38578	38583	38526	38726	38694	**38746**	**38746**
M15421(6)	**48052**	47669	47882	48048	48048	48048	**48052**	**48052**	**48052**
M15421(7)	**55171**	54891	55020	55048	55067	55072	55071	**55171**	**55171**
j02459(7)	**116700**	114381	116222	116257	115320	115301	116487	**116700**	**116700**
bx842596(4)	**227920**	224797	227252	226538	225782	223029	227238	**227920**	227914
bx842596(7)	**445422**	429338	443600	436739	438214	417680	442385	**445422**	444518
DNAgen									
acin1	**47618**	47264	47115	46955	46876	46865	47460	**47618**	47613
acin2	**151553**	147429	144133	144705	144634	144567	151353	151545	151456
acin3	**167877**	163965	156138	156630	156776	155789	167431	167861	167228
acin5	**163906**	161511	144541	146607	146591	145880	163313	163891	163426
acin7	**180966**	180052	155322	157984	158004	157032	180177	180924	180220
acin9	**344107**	335522	322768	324559	325930	314354	343315	344078	343385
f-series									
f25(305)	**596**	-	-	-	-	-	**596**	**596**	**596**
f25(400)	**777**	-	-	-	-	-	**777**	**777**	**777**
f25(500)	**921**	-	-	-	-	-	**921**	**921**	**921**
f50(315)	**1581**	-	-	-	-	-	1578	**1581**	**1581**
f50(412)	**1573**	-	-	-	-	-	1572	**1573**	**1573**
f50(498)	**1570**	-	-	-	-	-	**1570**	**1570**	**1570**
f100(307)	**2793**	-	-	-	-	-	2780	**2793**	**2793**
f100(415)	**2860**	-	-	-	-	-	2846	**2860**	**2860**
f100(512)	**2732**	-	-	-	-	-	2717	**2732**	2731
f508(354)	**18112**	-	-	-	-	-	17936	18110	18007
f635(350)	**22498**	-	-	-	-	-	22239	22492	22358
f737(355)	**25218**	-	-	-	-	-	24844	25206	24948
f1343(354)	**49042**	-	-	-	-	-	48205	49012	48121
f1577(354)	**57373**	-	-	-	-	-	56300	57313	56024

[15] Eugene W Myers, Granger G Sutton, Art L Delcher, Ian M Dew, Dan P Fasulo, Michael J Flanigan, Saul A Kravitz, Clark M Mobarry, Knut HJ Reinert, Karin A Remington, et al. A whole-genome assembly of drosophila. *Science*, 287(5461):2196–2204, 2000.

[16] Daniel R Zerbino and Ewan Birney. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome research*, 18(5):821–829, 2008.

[17] Jared T Simpson, Kim Wong, Shaun D Jackman, Jacqueline E Schein, Steven JM Jones, and Inanç Birol. Abyss: a parallel assembler for short read sequence data. *Genome research*, 19(6):1117–1123, 2009.

[18] Gabriel Luque and Enrique Alba. Metaheuristics for the dna fragment assembly problem. *International Journal of Computational Intelligence Research*, 1(1):98–108, 2005.

[19] Jacek Błażewicz, Piotr Formanowicz, Marta Kasprzak, Wojciech T Markiewicz, and J Wglarz. Tabu search for dna sequencing with false negatives and false positives. *European Journal of Operational Research*, 125(2):257–265, 2000.

[20] Enrique Alba and Gabriel Luque. A new local search algorithm for the dna fragment assembly problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 1–12. Springer, 2007.

[21] Abdelkamel Ben Ali, Gabriel Luque, Enrique Alba, and Kamal E Melkemi. An improved problem aware local search algorithm for the dna fragment assembly problem. *Soft Computing*, 21(7):1709–1720, 2017.

[22] Rebecca J Parsons, Stephanie Forrest, and Christian Burks. Genetic algorithms, operators, and dna fragment assembly. *Machine Learning*, 21(1-2):11–33, 1995.

[23] Enrique Alba and Gabriel Luque. A hybrid genetic algorithm for the dna fragment assembly problem. In *Recent advances in evolutionary computation for combinatorial optimization*, pages 101–112. Springer, 2008.

[24] Bernabé Dorronsoro, Enrique Alba, Gabriel Luque, and Pascal Bouvry. A self-adaptive cellular memetic algorithm for the dna fragment assembly problem. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2651–2658. IEEE, 2008.

[25] Gabriela Minetti, Guillermo Leguizamón, and Enrique Alba. An improved trajectory-based hybrid metaheuristic applied to the noisy dna fragment assembly problem. *Information Sciences*, 277:273–283, 2014.

[26] Ko-Wei Huang, Jui-Le Chen, and Chu-Sing Yang. A hybrid pso-based algorithm for solving dna fragment assembly problem. In *Innovations in Bio-Inspired Computing and Applications (IBICA), 2012 Third International Conference on*, pages 223–228. IEEE, 2012.

[27] Jesun Sahariar Firoz, M Sohel Rahman, and Tanay Kumar Saha. Bee algorithms for solving dna fragment assembly problem with noisy and noiseless data. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 201–208. ACM, 2012.

[28] Jesun Sahariar Firoz, M Sohel Rahman, and Tanay Kumar Saha. Hybrid meta-heuristics for dna fragment assembly problem for noiseless data. In *Informatics, Electronics & Vision (ICIEV), 2012 International Conference on*, pages 652–656. IEEE, 2012.

[29] Gabriela Minetti, Guillermo Leguizamón, and Enrique Alba. Sax: a new and efficient assembler for solving dna fragment assembly problem. In *2012 Argentine Symposium on Artificial Intelligence*, 2012.

[30] Ko-Wei Huang, Jui-Le Chen, Chu-Sing Yang, and Chun-Wei Tsai. A memetic particle swarm optimization algorithm for solving the dna fragment assembly problem. *Neural Computing and Applications*, 26(3):495–506, 2015.

[31] James Alexander Hughes, Sheridan Houghten, and Daniel Ashlock. Restarting and recentering genetic algorithm variations for dna fragment assembly: The necessity of a multi-strategy approach. *Biosystems*, 150:35–45, 2016.

[32] Mohcin Allaoui, Belaïd Ahiod, and Mohamed El Yafrani. A hybrid crow search algorithm for solving the dna fragment assembly problem. *Expert Systems with Applications*, 102:44–56, 2018.

[33] Guillermo M Mallén-Fullerton, James Alexander Hughes, Sheridan

Houghten, and Guillermo Fernández-Anaya. Benchmark datasets for the dna fragment assembly problem. *International Journal of Bio-Inspired Computation*, 5(6):384–394, 2013.

[34] Albert YS Lam and Victor OK Li. Chemical reaction optimization: A tutorial. *Memetic Computing*, 4(1):3–17, 2012.

[35] Timothy Starkweather, S McDaniel, Keith E Mathias, L Darrell Whitley, and C Whitley. A comparison of genetic sequencing operators. In *ICGA*, pages 69–76, 1991.

[36] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[37] F Yu Vincent, AAN Perwira Redi, Yosi Agustina Hidayat, and Oktaviyanto Jimat Wibowo. A simulated annealing heuristic for the hybrid vehicle routing problem. *Applied Soft Computing*, 53:119–132, 2017.

[38] Nader Ghaffarinasab, Alireza Motallebzadeh, Younis Jabarzadeh, and Bahar Y Kara. Efficient simulated annealing based solution approaches to the competitive single and multiple allocation hub location problems. *Computers & Operations Research*, 90:173–192, 2018.

[39] Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.

[40] Gabriela Minetti and Enrique Alba. Metaheuristic assemblers of dna strands: Noiseless and noisy cases. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.