Expert Systems With Applications 102 (2018) 44–56

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A hybrid crow search algorithm for solving the DNA fragment assembly problem

Mohcin Allaoui*, Belaïd Ahiod, Mohamed El Yafrani

LRIT, Associated Unit to CNRST (URAC 29), Faculty of Science, Mohammed V University in Rabat, Rabat B.P. 1014, Morocco

ARTICLE INFO

Article history:
Received 20 July 2017
Revised 19 January 2018
Accepted 8 February 2018
Available online 15 February 2018

Keywords:
Crow search algorithm
DNA fragment assembly
Discrete optimization
PALS2-many*

ABSTRACT

The sequencing of DNA goes through a step of fragment assembly, this step is known as DNA fragment assembly problem (FAP). Fragment assembly is considered as an *NP*-hard problem, which means there is no known polynomial-time exact approach, hence the need for meta-heuristics. Three major strategies are widely used to tackle this problem: greedy graph-based algorithms, de Bruijn graphs, and the overlay-layout-consensus (OLC) approach. In this paper, we propose an adaptation of the novel crow search algorithm (CSA) to solve the DNA fragment assembly problem following the OLC model. In order to accelerate the search process and improve the quality of the solutions, we combined CSA with a local search method. Using this combination we were able to obtain very accurate solutions for all the instances of the DNA fragment assembly problem we tested. In fact, our algorithm outperformed all other algorithms designed for the same purpose. Our contribution consists in the implementation of a new assembler for the DNA fragment assembly problem capable of finding for the first time the optimal solutions for 20 out of 30 instances. The approach we proposed to adapt CSA for a discrete optimization problem is a novelty. We preserved the semantics of the original algorithm by applying standard operators from evolutionary algorithms. Following the same approach can make adapting new algorithms for discrete problems more accessible and more efficient compared to mapping algorithms designed for continuous optimization to combinatorial problems.

1. Introduction

In bio-informatics manipulating DNA sequences is a frequent task (e.g.: multiple sequence alignment, finding DNA sequence motifs). The current sequencing technologies allow to read sequences with lengths varying between 20 to 1000 bases. However the human genome is about 3.2 billion bases. Therefore, in order to determine long DNA sequences, the shotgun sequencing is used to overcome this limitation. The idea behind shotgun sequencing is to cut a long sequence of DNA into many pieces called fragments and then to use an assembler to (hopefully) find the original sequence. This is done by making multiple copies of the same DNA sequence, such that each copy is broken into many short fragments, causing the loss of the original order of the fragments. A subset of the fragments is read using Sanger sequencing, and then the length of the overlap between each couple of fragments is calculated. Afterwards, the assembler searches for the arrangement of the fragments that maximizes the sum of all the overlaps of each adjacent fragments. Finally, a multiple sequence aligner is applied to build the complete sequence. This is referred to as the overlap-layout-consensus approach (OLC). The layout phase, in which the order of the fragments is determinated is an *NP*-hard problem (Pevzner, 2000).

The OLC approach is an excellent way for assembling reads obtained by Sanger sequencing. However, the newest sequencing technologies, such as, next generation sequencing (NGS) produce a high volume of much smaller sequence fragments. Greedy graph-based assemblers and de Bruijn graph strategies became popular for such technologies (Miller, Koren, & Sutton, 2010). Even though the NGS technologies produce short fragments, the OLC approach can still be used for such technologies. In fact, using the OLC approach multiple large eukaryotic genomes have been successfully assembled from short reads (Chinwalla et al., 2002). Additionally, newer NGS technologies have been improved to produce long reads, and thus allowing the OLC approach to become favorable once again (Earl et al., 2011).

Due to its importance, a lot of work has been done to solve the DNA fragment assembly. However, the field is still wide open for research. Indeed, fast and more accurate assemblers are a necessity. The commonly used assemblers are based on greedy

* Corresponding author.
E-mail addresses: mohcin.allaoui@um5s.net.ma (M. Allaoui), ahiod@fsr.ac.ma (B. Ahiod), melyafrani@acm.org (M. El Yafrani).

https://doi.org/10.1016/j.eswa.2018.02.018

Table 1
A comparison of some state-of-the-art approaches for solving FAP.

Algorithm	Description	Pros	Cons
T/D RGGA (Hughes et al., 2016)	A set of multiple variants of the restarting and recentring genetic algorithm.	The ability of the RRGA design to escape local optima lead to high quality solutions.	Choosing the best variant for a specific instance is difficult due to the high number of variants provided.
PPSO+DE (Mallén-Fullerton & Fernandez-Anaya, 2013)	A parallel hybrid algorithm between Particle Swarm Optimization (PSO) and Differential Evolution (DE).	The technique used to eliminate the redundant fragment reduces the size of the instance, which can reduce its complexity.	The algorithm was enable to find good solutions for the medium size instances.
QEGA (Firoz et al., 2012)	Queen bee evolution based on genetic algorithm.	The approach used was well optimized for noisy data.	An inferior performance for the noisy data.
SA (Minetti & Alba, 2010)	The simulated annealing algorithm for FAP.	Very efficient approach for small instances.	When applied for noisy instances it showed an inferior performance.
PALS (Alba & Luque, 2007)	A problem aware local search.	Very fast heuristic without any parameters to tune.	The heuristic falls quickly into local optima without providing a strategy to escape it.

techniques, such as, PHRAP (Green, 1994), TIGER (Sutton, White, Adams, & Kerlavage, 1995), and CAP3 (Huang & Madan, 1999). These are full-featured packages that can handle the three phases of the OLC approach. Most published work consists in solving only the layout phase. Evolutionary Algorithms (EA) were studied extensively, especially Genetic Algorithms (GA), as in Parsons, Forrest, and Burks (1995), where traditional permutation and sorted-order representations with their associated operators are compared. The edge-recombination crossover found to perform best. In Nebro, Luque, Luna, and Alba (2008) a grid-based GA was proposed, which uses a panmictic population, and it is based on computing parallel function evaluations in an asynchronous way. Heuristics such as Simulated Annealing (SA) was used as stand alone assembler in Minetti and Alba (2010). It was successfully applied on both noisy and noiseless data. SA was also used as an operator inside a Particle Swarm Optimization algorithm (PSO) (Huang, Chen, & Yang, 2012). The smallest position value (SPV) rule was used for encoding the particles in order to enable PSO to be suitable for solving FAP. A particular heuristic called PALS (Problem Aware Local Search) (Alba & Luque, 2007) was able to find accurate solutions, and was improved in Ali, Luque, Alba, and Melkemi (2015) to obtain even better results in significantly short times. Nature-inspired meta-heuristics were also adapted to solve FAP, including Ant Colony Optimization (ACO) (Meksangsouy & Chaiyaratana, 2003), in which an asymmetric ordering representation is proposed. A path cooperatively generated by all ants in the colony represents the search solution. Queen-bee Evaluation based on Genetic Algorithm (QEGA) (Firoz, Rahman, & Saha, 2012) was adapted using PALS for the modification of the permutation under consideration. Another nature inspired algorithm based on Particle Swarm Optimization and Differential Evolution (PPSO+DE) was presented in Mallén-Fullerton and Fernandez-Anaya (2013). The most recent work related to this topic is the use of multiple variants of Genetic Algorithm: Recentering–Restarting GA (T/D RRGA), Island Model GA (IM), and a GA which uses Ring Species (RS) (Hughes, Houghten, & Ashlock, 2016). The idea was to use these variations in combination to better explore the search space.

We summarize in Table 1 a comparison of some state-of-the-art approaches for solving FAP. We picked the algorithms that use the benchmark instances proposed in Mallén-Fullerton, Hughes, Houghten, and Fernández-Anaya (2013).

This paper presents an adaptation of the crow search algorithm (CSA) (Askarzadeh, 2016), for solving the DNA fragment assembly problem. CSA is a novel meta-heuristic inspired by the intelligence of crows. It is based on their foraging behavior. CSA is shown to be very effective in solving continuous optimization problems, such as pressure vessel design problem and gear train design problem (Askarzadeh, 2016). Due to the fact that CSA is a population based meta-heuristic, it is characterized by high diversification abilities. Therefore, it is combined with a local search method based on the improved PALS (Ali et al., 2015), in order to improve its capability to exploit particular areas in the search space.

The methods we proposed in this paper are based on the fitness of the solutions to guide the search process, this can be a drawback since the ultimate goal is to minimize the number of contigs. To our knowledge, no population based method is proposed where the number of contigs is the search guide, and we believe if such method exists it would lead to better results.

The rest of this paper is organized as follows: in Section 2 we rigorously describe the DNA assembly problem. Section 3 is an overview of the original algorithm CSA. Section 4 describes our proposition to solve FAP using CSA. The results of our numerical experiments on three sets of frequently used benchmarks are presented in detail in Section 5 as well as parameter tuning and performance analysis. A conclusion is presented in Section 6.

2. DNA fragment assembly problem

The goal of the DNA fragment assembly is the reconstruction of a long DNA strand starting from a set of random fragments generated using the shotgun sequencing. Based on the OLC model, this process is achieved using the following steps.

- Overlay: The first step consists in calculating the overlaps between all the fragments. In order to accomplish this, we compare each pair of fragments to obtain the largest overlap (match) between the two fragments. This is done mostly using dynamic programming applied to semi-global alignments such as Smith–Waterman algorithm (Smith & Waterman, 1981).
- Layout: This is the most difficult step of the OLC approach. The goal is to find the correct position of each fragment, by searching for the ordering of the fragments that maximize the sum of the overlap score of each consecutive fragments. This step is proven to be an *NP*-hard problem (Pevzner, 2000).
- Consensus: The final step is the reconstruction of the complete DNA sequence based on the layout generated in the previous steps. This task is relatively easy since we apply a simple majority rule.

An illustration of these steps is shown in Fig. 1. In this paper, we only focus on the layout phase, since it constitutes the core of the DNA fragment assembly problem. The algorithms we used and the results we obtained are presented in the following sections.

3. Crow search algorithm

3.1. Basic CSA

Crows are one of the most intelligent creatures. Many of their behaviors prove this high level of cleverness, such as, tool making