Contents lists available at ScienceDirect

Chemical Engineering Research and Design

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

A note on short-term scheduling of multi-grade polymer plant using DNA computing

Deepak Sharma, Manojkumar Ramteke[*]

Department of Chemical Engineering, Indian Institute of Technology Delhi, 110016, India

ARTICLE INFO

Article history:
Received 21 August 2017
Received in revised form 10 March 2018
Accepted 16 May 2018
Available online 26 May 2018

Keywords:
DNA computing
Short-term scheduling
Mathematical programming
Batch-plant

ABSTRACT

Short-term scheduling of batch polymer plant involves the scheduling of different orders in parallel available production lines. The scheduling becomes more challenging due to the presence of sequence-dependent changeover constraints between different orders which lead to combinatorial optimization formulation. Such combinatorial optimization problems have exponential time complexity on the silicon-based computer. DNA computing experiments are found to be promising for such combinatorial optimization problems particularly involving unique feasible optimal solution. However, use of DNA to find a solution to real-life problems involving multiple feasible solutions is an emerging area of research. The present paper illustrates the DNA solutions to the short-term scheduling of a polymer plant involving multiple feasible solutions and parallel production lines. The DNA computer aided with nearest neighbour heuristics and iterative implementation found to be successfully searching the optimal solution in a combinatorial search space for three short-term scheduling problems of multi-grade polymer plant.

1. Introduction

In a polymer industry, there are several polymer grades which are produced according to their uses. This difference in grades is mainly because of the color, brightness or mechanical strength. This difference depends on the operating conditions of the polymer plant which is achieved by varying monomer conversion, molecular weight distribution, and external coloring. Therefore, by changing the operating conditions in a sequential manner one can produce different grades of the same polymer. Short-term scheduling is to find a solution as a sequence of orders to be produced for optimal operation. In this, the details of each order such as product grade, amount, release time and due time are known a priory. The order due time shows the interval at which it should be delivered otherwise it will be considered as late, and release time is the earliest starting time of processing of an order.

Scheduling is a combinatorial optimization problem. Popularly, the scheduling optimization is performed using mathematical programming in which the problem is formulated as a discrete-time or continuous time (Floudas and Lin, 2004; Kondili et al., 1993). The mathematical programming methods are commonly used for short-term scheduling optimization and are reviewed by Mendez et al. (2000). Recently several studies are reported in the literature on short-term scheduling optimization of batch plants (Kotecha et al., 2008; Shaik and Bhat, 2013; Shaik and Vooradi, 2013; Vooradi and Shaik, 2012) using mathematical programming.

Ramteke and Srinivasan (2011) studied a short-term scheduling in multiproduct polymer plant using the real-coded genetic algorithm. The case study they had used involved processing of twenty different product orders in four extruders where each product order required a different processing time depending on the extruder used for processing. In addition, the processing involved sequence dependent changeover times and forbidden sequences of product orders. This led to a combinatorial optimization formulation. DNA computing has become popular in the current era to solve such complex combinatorial problems. The DNA computing shows massive parallelism, unique data structure, and high information density than silicon-based computers. Silicon-based computers usually perform the calculations in a sequential manner. However, DNA computing operates in a parallel manner and thus allows us to go through a computationally intensive problem and look into it from a different point of view.

Miniaturization of transistors to improve the computational speed is not possible in future as the size of transistors is already approaching

[*] *Corresponding author.*
E-mail address: mcramteke@chemical.iitd.ac.in (M. Ramteke).

https://doi.org/10.1016/j.cherd.2018.05.021
0263-8762/

to a molecular level (Gartenberg, 2016) and designing these at such a small scale is more difficult due to quantum effects. Therefore, the development of alternate computing ways using molecules is a need for the future. Molecular computers, in future, might show a proficiency in solving problems which are difficult to solve using conventional methods. The outstanding energy efficiency, massive parallelism, and amazing data storage characteristic in molecular computation has raised the opportunity of computation at the molecular level. The first such illustration is a DNA computer in which the massive parallelism of DNA molecules is used intelligently to counter the intractable nature of mathematical problems. DNA computer generates all feasible solutions for such problems in just one biochemical step in contrast with a silicon-based computer where generating these requires exponential time complexity. An optimal solution is then extracted from all feasible solutions using different biochemical techniques such as affinity separation, gel electrophoresis, and polymerase chain reaction (PCR) etc.

In this study, the DNA computing procedure is extended to solve real-life grade scheduling problems. For this, three polymer grade scheduling problems were formulated from the existing problems in the literature (Ramteke and Srinivasan, 2011) and are discussed in detail in Section 3. The first problem involves the unique feasible solution and leads to the Hamiltonian Path Problem (HPP) comprising seven nodes. This problem is solved using the traditional DNA computing procedure. The DNA computing procedure is then extended by incorporating the nearest neighbour heuristics (Rego et al., 2011) to solve the problems involving multiple feasible solutions. This is illustrated by solving a second problem which has a single production line and involves multiple feasible solutions. The solution procedure is further extended by incorporating the iterative implementation to solve more realistic problems having multiple production lines and multiple feasible solutions. The developed methodology is illustrated on a third problem which has four parallel production lines and involves multiple feasible solutions.

2. DNA computing

DNA (Deoxyribonucleic acid) stores the genetic information of almost all living organisms. This information is stored using four nucleotide bases thymine (T), cytosine (C), adenine (A), and guanine (G). DNA comprises two long chains of nucleotides which intertwine to form a double helix structure. The nucleotide chain (strand) is formed by nucleotides joined together by a phosphodiester bond. Two such complementary strands of nucleotides attach to each other by hydrogen bonds to form a helical structure which looks like a spiral staircase.

The advances in biochemical techniques over the years to synthesize, extract, modify and multiply the DNA molecules has made it easy to use DNA molecules for novel applications such as computing. Such application of DNA computing is illustrated for the first time by Leonard Adleman (Adleman, 1994) in 1994. In his experiment, he used DNA sequences to represent vertices and edges of the directed graph and operated these using commonly used biochemical techniques to find a Hamiltonian path, a route that visits each vertex exactly once. Following were the steps used by Adleman (1994) to solve HPP:

- Step 1: Generate random path in a given graph.
- Step 2: Starting and ending vertices should be vertex (V_i) and vertex (V_f).
- Step 3: From n vertex graph, keep routes that visit just n vertices.
- Step 4: Path visiting every vertex at least and only once is retained.
- Step 5: If path found, say “Yes”; or else, “No”.

Adleman performed molecular biology experiments to solve HPP for a given graph structure having seven vertices and thirteen edges. In these experiments, he encoded the DNA sequences for all vertices and edges. Next, he mixed these sequences for ligation where these sequences joined in all possible combinations. Thereafter, he operated on these DNA molecules of different lengths and combinations using filtration steps such as gel electrophoresis, PCR and affinity separation in which all DNA strands of incorrect length and strands not having all required vertices were removed. The remaining strands after filtration showed the solution to the HPP.

After Adleman, several researchers have extended DNA computing for solving challenging mathematical problems. Lipton (1995) solved 2-variable SAT (Boolean Satisfiability) problem using DNA computing. Guarnieri et al. (1996) developed a DNA-based algorithm for the addition operation. Ouyang et al. (1997) solved the six-vertex Maximal Clique Problem using DNA computing. Braich et al. (2002) extended the DNA computing for solving 20 variables 3-SAT problem. Liu et al. (2000) solved an instance of four variable SAT using surface assisted DNA computing. Faulhammer et al. (2000) combined DNA and RNA molecules together in a molecular computing to solve 9-variable SAT problem. Sakamoto et al. (2000) used single-stranded DNA for molecular computation by exploiting the hairpin formation to solve the SAT problem. These experiments showed that solving the problem with DNA gives a high parallelism over silicon-based computers; however, there are some obstacles in the way of DNA computing such as algorithm speed, real-time updating, etc.

DNA computing is also applied to the problems involving variation in objective values. Narayanan and Zorbalas (1998) purposed an algorithm of proportional length to solve travelling salesman problem (TSP). In this study, they have additionally used varying concentrations of the DNA molecules to solve the TSP where variation in concentration represented the cost of travelling from one city to another city. This method demonstrated that the adaptation of local search is also feasible in DNA computing. Lee et al. (2004) and Xikui and Yan (2009) further improved this technique by including a temperature gradient as the cost of the travelling. Ibrahim et al. (2007) proposed a new hybrid method based on the method developed by Narayanan and Zorbalas (1998) in which they combined the variation in concentration and length of DNA sequences simultaneously. In this, the concentration controlled the hybridization of DNA molecules whereas the length controlled the cost of travelling. Even with these advances, the largest problem solved till date using DNA computing is a TSP with 15 vertices and 210 edges by Xiong et al. (2009). A large problem like this requires high accuracy and efficiency.

Recently, DNA origami (Rothemund, 2005) is used to solve the optimization problems. In DNA origami, complementary base pair property is used to form 2D or 3D structures (Veneziano et al., 2016) in which large size single-stranded DNA is folded into a desired shape by specific small size complementary DNA strands. Also, the attempts have been made in the literature to incorporate the programming language in the biological systems (Nielsen et al., 2016; Yang et al., 2016). Further, the DNA structures are now being analysed for suitability to store the data. Researchers have successfully stored the images (Bornholt et al., 2016) and large data (Bonnet et al., 2012; Church et al., 2012; Laddha, 2016; Molecule and Storing, 2016; Shah et al., 2013) on DNA. So far, up to 220 MB data is successfully stored using DNA (Rosenblum, 2016).