An Efficient Algorithm for Community Detection in Complex Weighted Networks

Leila Samandari Masooleh[1], Jeffrey E. Arbogast[2,3], Warren D. Seider[4], Ulku Oktem[5], and Masoud Soroush[1*]

[1]Department of Chemical and Biological Engineering, Drexel University, Philadelphia, PA 19104, USA

[2]American Air Liquide, Newark, DE 19702, USA

[3]Air Liquide (China) R&D Co., Ltd., Shanghai, China 201108

[4]Department of Chemical and Biomolecular Engineering, University of Pennsylvania, Philadelphia, PA 19104-6393, USA

[5]Near-Miss Management, LLC, 1800 JFK Blvd., Suite 300, Philadelphia, PA 19103, USA

July 31, 2020

Submitted for Publication in *AIChE Journal*

Keywords: Complex network, Community detection, Whale optimization algorithm, Multi-objective optimization

[*]Corresponding author. soroushm@drexel.edu; (215) 895-1710 (phone); (215) 895-5837 (fax)

Abstract

Community detection decomposes large-scale, complex networks ‘optimally’ into sets of smaller sub-networks. It finds sub-networks that have the least inter-connections and the most intra-connections. This article presents an efficient community detection algorithm that detects community structures in a weighted network by solving a multi-objective optimization problem. The whale optimization algorithm is extended to enabe it to handle multi-objective optimization problems with discrete variables and to solve the problems on parallel processors. To this end, the population’s positions are discretized using a transfer function that maps real variables to discrete variables, the initialization steps for the algorithm are modified to prevent generating unrealistic connections between variables, and the updating step of the algorithm is redefined to produce integer numbers. To identify the community configurations that are Pareto optimal, the non-dominated sorting concept is adopted. The proposed algorithm is tested on the Tennessee Eastman process to show its application and performance.

1. Introduction

Real-world networks are complex and large scale, represent very interacting entities, and generate diverse types of data. Biological networks [1], power networks [2], chemical process control [3-6], and many others [7-9] are some examples of complex networks. Moreover, modern manufacturing plants are increasingly integrated [10], leading to structural and computational complexities. In recent years the problem of decomposing a large complex network into a set of interacting small networks that adequately capture the interactions of the original large network has received great attention in many engineering and science fields[11-14].

An important problem in complex networks is to detect communities accurately [1]. Consider a subset of nodes (variables) in a network. If the variables in this subset have more interactions with each other than with the rest of the network, then the subset is called a community. The process of identifying these communities is called community detection. Recently, many powerful methods have been introduced for community detection in large-scale networks. A comprehensive review of existing methods can be found in Ref. [15]. Among these methods, modularity optimization has been used most for detecting communities in complex networks [16].

Community detection based on optimizing a modularity function is considered as an NP-hard problem [17]. To solve this optimization problem, different metaheuristic algorithms, from evolutionary-based algorithms to swarm-based algorithms, have been adopted [12, 18-22]. The algorithms optimize a single-objective or a multi-objective function [12]. Obviously, the computational efficiency of each community detection method depends on the optimization problem solver that the community detection method uses.

Single-objective community detection methods have received significant attention. Tasgin et al. [22] employed the genetic algorithm (GA) to optimize modularity in complex networks. Unlike traditional methods like partitional clustering, their approach does not require the number of communities in advance [22]. Cai et al. [18] adopted a particle swarm optimization (PSO) algorithm to partition a signed network (i.e., a network with both positive and negative links). Differential evolution (DE) [20], a hybrid algorithm based on PSO and extremal optimization [21], a discrete bat algorithm (BA) [19], artificial bee colony swarm optimization [23], and several other metaheuristic algorithms have been used in community detection. A comparative analysis of metaheuristic algorithms in community detection can be found in Ref. [24, 25].

The inability of single-objective community detection methods in detecting multiple potential communities has motivated the use of multi-objective community detection methods [26]. A multi-objective community detection method typically considers two objective functions, one is used to maximize the intra-connection, and the other to minimize the inter-connection. A Pareto dominance approach [27, 28] or a decomposition approach [29-31] has mostly been used to solve the multi-objective optimization problems.

Apart from choosing an appropriate optimization problem solver, creating a precise graph which models and indicates the relation and intensity between different nodes in the network, is important in identifying communities. In graph theory, the impact of variables on each other is reflected by setting a weight on each edge of the graph. Studies have shown that the strength of the interaction in real networks is an essential parameter in partitioning, especially for dense networks [32]. Depending on the application, there are different approaches for evaluating edge weights, which may be practical or impractical depending on the specific case under consideration [1, 33].

As computational problems involved in detecting communities in the real-world networks are nonconvex and large scale, the computational efficiency of methods applied to these problems is of importance. This points to the importance of developing and implementing parallel algorithms. Among existing community detection methods, only a few have been implemented in parallel [34, 35].

In this article, we propose a multi-objective community detection algorithm that uses the concept of community structure [1] to decompose a large-scale network into multiple sub-networks. We formulate community detection as a multi-objective optimization problem. We employ the concept of weighted modularity [36] to conduct partitioning and identify the communities. The strength of the interaction between each pair of nodes is determined by conducting sensitivity analyses. Given that WOA has been shown to be suitable for optimizing the composite mathematical functions (those with local optima) due to the simultaneous balance between exploration and exploitation phase, we extend the WOA to solve community detection multi-objective optimization problems with discrete variables. The extension involves: redefining the population's position vectors in a discrete form by using a transfer function that maps real variables to discrete variables; modifying the initialization steps for the algorithm to prevent generating unrealistic connections between variables; and redefining the updating step of the algorithm to

produce integer numbers. We use the non-sorting genetic algorithm (NSGA)-II [27] to generate a list of non-dominated solutions. To the best of our knowledge, this work is the first attempt to solve community detection as a multi-objective optimization problem using discrete WOA and apply the algorithm to complex network clustering like process industry problems. We use the inherently parallel nature of WOA to address the slow speed of the convergence of the existing community detection algorithms, especially for large-scale networks. The proposed algorithm is applied to the Tennessee Eastman which has been used extensively as a benchmark in chemical engineering.

The article proceeds as follows. Section 2 reviews some necessary preliminaries relevant to this work. Section 3 describes the proposed community detection method. Section 4 shows the implementation of the proposed method on the Tennessee Eastman process. The article ends with concluding remarks.

2. Preliminaries

Before describing our proposed method, we briefly review some necessary basic terminologies and concepts in community detection.

2.1. Community Detection

A network can be presented by a graph $G = (V, E)$, where V is a set of nodes (also called vertices), and E is a set of links (also called edges).

Definition 1 [37]. Each graph can be described by an $n \times n$ matrix, called adjacency matrix, where n stands for the number of nodes of the graph. The ijth element of an adjacency matrix A in a directed graph, A_{ij}, takes a non-zero real value, if the *i*th variable affects the *j*th variable. It is set to zero, if the *i*th variable does not affect the *j*th variable. Note that a variable can affect itself. In an unweighted graph, each of these elements takes a value of 1 or 0.

Definition 2 [37]. The degree of a node i, dented by k_i, is the number of links associated with the node i. In a directed graph, every node has an in-degree and an out-degree. In directed graphs, the in-degree of a node i, denoted by k_i^{in}, is the number of links directed towards the node i (number of variables that affect the variable i): $k_i^{in} = \sum_{j \in V} A_{ji}$, and the out-degree of a node i, denoted by k_i^{out}, is the number of links directed from node i (number of variables that are affected by the variable i): $k_i^{out} = \sum_{j \in V} A_{ij}$.

Definition 3 [37]**.** A community is a group of nodes whose intra-connections are stronger than their inter-connections. For a sub-network S, $S \subset V$, the number of links connecting node i in the sub-network S to other nodes in the same sub-network, $k_{i,S}^{in} = \sum_{j \in S} A_{ij}$. The number of links connecting node i in the sub-network S to the nodes that do not belong to S, $k_{i,S}^{out} = \sum_{j \notin S} A_{ij}$. According to the definition of a community, a subnetwork S is a "community in a strong sense", if $\forall\, i \in S,\ k_{i,S}^{in} > k_{i,S}^{out}$, and it is a "community in a weak sense", if $\sum_{i \in S} k_{i,S}^{in} > \sum_{i \in S} k_{i,S}^{out}$.

2.2. Multi-objective Optimization

A multi-objective optimization problem involves several (often conflicting) objective functions that should be optimized simultaneously. Multi-objective optimization allows for analyzing tradeoffs among competing objectives and generating a set of solutions. It may have no single solution that optimizes all objective functions; as one objective improves, another deteriorates. A multi-objective optimization problem can be represented by $\min_z F(z) = \{f_1(z), \cdots, f_p(z)\}$, where $z \in Z \subset R^{n_z}$ is the vector of decision variables, and $f_1(z), \cdots, f_p(z)$ are objective functions.

Definition 4 [38]**.** In a minimization problem, a decision vector $z_A \in Z$ is said to dominate a decision vector $z_B \in Z$, if for every $i \in \{1, \cdots, p\}$, $f_i\ (z_A) \leq f_i\ (z_B)$ and if there is a $j \in \{1,2, \cdots, p\}$, $f_j\ (z_A) < f_j\ (z_B)$.

Definition 5 [38]**.** A decision vector $z^* \in Z$ is the Pareto optimal or nondominated solution, if it is not dominated by any other solutions.

Definition 6 [39]**.** A Pareto front is the set of all nondominated solutions at which no objective function can be improved without sacrificing another objective function.

2.3. Modularity Function

Modularity has been used widely in graph partitioning. It is defined as "the difference between the fraction of edges within communities in the network and the expected fraction of such edges, which are randomly distributed" [16]. Mathematically, modularity for directed and unweighted graphs is defined as [16]:

$$Q_{dir} = \frac{1}{m} \sum_i \sum_j \left[A_{ij} - \frac{k_i^{in} k_j^{out}}{m} \right] \delta(c_i, c_j) \quad (1)$$

where $m = \sum_{i \in V} k_i^{in} = \sum_{i \in V} k_i^{out} = \sum_i \sum_j A_{ij}$ stands for the total number of links in the network, c_i is the community of node i, and δ is the Kronecker delta symbol. If both i and j belong to the same community, δ is equal to 1; otherwise, it is zero. The definition implies that a good partition has a large modularity value.

2.4. Whale Optimization Algorithm

The WOA is a swarm-based metaheuristic algorithm inspired by the hunting behavior of humpback whales [40]. Humpback whales have a particular method in hunting named bubble-net feeding [41], which hunt a group of small fish on the surface by making unique bubbles "along a circular path or '9' shaped path". Like other population-based metaheuristic optimization algorithms, the WOA has an exploration and an exploitation phase. To formulate this method mathematically, the hunting mechanism of humpback whales was divided into three phases: shrinking encircling prey, spiral updating, and searching for prey. The first two phases handle the exploitation phase, and the last phase does the search.

2.4.1. Encircling Prey

The algorithm initially assigns random positions (of dimension n_z) to θ whales (θ is set by the user) and evaluates the cost function at the θ positions. n_z is the number of decision variables in the optimization problem. It compares these cost function values and chooses the whale, whose position has the lowest cost function value (in a minimization problem), as the leader. It then updates the positions of the whales, $z^1, \cdots, z^\theta$, based on the position of the leader whale, using:

$$z^j(t+1) = z^*(t) - A^j D^j \tag{2}$$

where $j = 1, \cdots, \theta$, $z^j(t)$ is the position of the jth whale in the current iteration, $z^*(t)$ is the position of the leader whale in the current iteration, t represents the current iterations.

$$D^j = \left[\left| C^j z_1^*(t) - z_1^j(t) \right| \ \cdots \ \left| C^j z_{n_z}^*(t) - z_{n_z}^j(t) \right| \right]^T \tag{3}$$

$$A^j = 2\, a\, r_1^j - a \tag{4}$$

$$C^j = 2\, r_2^j \tag{5}$$

Here, $|x|$ is the absolute value of a scalar x, a^j linearly decreases from 2 to 0 with the iterations according to:

$$a = 2\left(1 - \frac{t}{t_{max}}\right) \tag{6}$$

to keep a balance between exploration and exploitation search, and r_1^j and r_2^j are random variables with uniform distributions in the interval $[0, 1]$. t_{max} is the maximum number of iterations. Once the positions of the whales in the next iteration, $z^1(t+1), \cdots, z^\theta(t+1)$, are calculated, the whale position at which, the cost function has the lowest value is chosen as $z^*(t+1)$, which is the position of the leader whale in the next iteration.

2.4.2. Bubble-net Attacking

To describe hunting by humpback whales mathematically, the WOA considers two mechanisms called shrinking encircling and spiral updating. An equal probability is assigned to each mechanism to update the position of individual whales in each iteration according to:

$$z^j(t+1) = \begin{cases} z^*(t) - A^j D^j & \text{if } P^j < 0.5 \quad (a) \\ \bar{D}^j e^{bl} \cos(2\pi l) + z^*(t) & \text{if } P^j \geq 0.5 \quad (b) \end{cases} \tag{7}$$

where

$$\bar{D}^j = \left[\left|z_1^*(t) - z_1^j(t)\right| \cdots \left|z_{n_z}^*(t) - z_{n_z}^j(t)\right|\right]^T,$$

P^j is a random variable with a uniform distribution in the interval $[0, 1]$, which models the shrinking encircling and spiral updating mechanism for the jth whale. b (usually set to 1) is a constant number that specifies the shape of the logarithmic spiral, and l is a random variable with a uniform distribution in the interval $[-1, 1]$. When $P^j < 0.5$, the decrease of a with each iteration allows for modeling the shrinking encircling mechanism. When $P^j \geq 0.5$, the expression that includes the cosine function, models the helix-shaped movement of whales.

2.4.3. Searching for Prey

This step is considered as an exploration stage aimed to expand the search by forcing the whales to move away from each other. In fact, this step attempts to achieve good coverage of the whole search space. The exploration phase is carried out using:

$$z^j(t+1) = \tilde{z}(t) - A^j \bar{\bar{D}}^j \tag{8}$$

$$\bar{\bar{D}}^j = \left[\left|C^j \tilde{z}_1(t) - z_1^j(t)\right| \cdots \left|C^j \tilde{z}_{n_z}(t) - z_{n_z}^j(t)\right|\right]^T \tag{9}$$

where $\tilde{z}(t)$ is selected randomly from the pool of the current positions of the whales.

2.4.4. Implementation of the Whale Optimization Algorithm

The WOA starts with randomly positioned whales. The position of jth whale is updated using Eq.(7) when $|A^j| \leq 1$ and Eq.(8) when $|A^j| > 1$. The WOA ends when a termination condition set by the user is satisfied.

3. Community Detection Method

This section describes our algorithm for community detection in process industries. Decomposing large-scale plants into small sub-systems is of interest for distributed control, real-time optimization, process synthesis and design, state estimation, and model-predictive safety, among others. To develop a community detection method suitable for these applications, community detection is formulated as a multi-objective optimization problem based on a state-space model. The multi-objective formulation allows for generating multiple configurations in the order of their modularity amounts. The availability of multiple configurations provides the user with multiple choices to select from. For example, in the distributed control application, sub-systems with maximum modularity may not pass the controllability test, and in distributed state estimation, sub-systems with maximum modularity may not pass the observability test. In these cases, the multi-objective optimization formulation allows for selecting a configuration that has adequate observability/detectability and high modularity. Unlike commonly used methods such as partitional clustering and hierarchical clustering which become computationally expensive when the dimension of the system increases, the proposed algorithm takes advantage of the fast convergence and parallelization feature of WOA. As the original version of WOA was designed for single-objective optimization problems with continuous variables, we modify the original version of WOA to make it suitable for handling multi-objective optimization problems with discrete variables. We also adopt the non-dominated sorting algorithm of NSGA-II to generate non-dominated set of solutions.

3.1. Objective Function

In many real-world networks, the connection between each pair of nodes (variables) is not binary (i.e., a link between two nodes is either present or not). Furthermore, a binary connection does not provide useful information about a network and may affect the optimal topology in community detection. To address this, the strength (sensitivity) of each link (a variable to another

variable) is accounted for in the form of a weight on the link. Hence, the modularity for directed networks, described in Eq.(1), was generalized to weighted modularity, Q_w, given by [36]:

$$Q_w = Q_1 - Q_2 \tag{10}$$

where

$$Q_1 = \frac{1}{W}\sum_i\sum_j w_{ij}\ \delta(c_i, c_j), \qquad Q_2 = \frac{1}{W}\sum_i\sum_j \frac{w_i^{out} w_j^{in}}{W}\ \delta(c_i, c_j)$$

w_{ij} is the *ij*th element of weighted adjacency matrix, $w_i^{out} = \sum_j w_{ij}$, $w_j^{in} = \sum_i w_{ij}$, and W = $\sum_i w_i^{out} = \sum_j w_j^{in} = \sum_i \sum_j w_{ij}$. The goal of optimization in community detection is to maximize Q_w, which consists of two conflicting objective functions, Q_1 and Q_2. Maximizing Q_w requires maximizing intra-connections in every sub-graph (Q_1) and minimizing the inter-connections between sub-graphs (Q_2).

3.2. Problem Formulation and Development of Weighted System Digraph

To identify communities in a network, the digraph of the entire network should be constructed. Decomposition methods based on graph theory typically use a state-space model of the network under consideration:

$$\begin{cases} x(k+1) = g(x(k), u(k)) \\ \quad y(k) = h(x(k)) \end{cases} \tag{11}$$

where $x \in \mathbb{R}^{n_x}$ is the vector of state variables, $u \in \mathbb{R}^{n_u}$ is the vector of input variables, $y \in \mathbb{R}^{n_y}$ is the vector of output variables, and g and h are smooth vector functions. The variables are assumed to be normalized; they are dimensionless and take a value in [0, 1]. For example, $x_i = (\mu_i - \mu_{i,min})/(\mu_{i,max} - \mu_{i,min})$, where μ_i is the original state variable with a dimension, $\mu_{i,min}$ is the lowest value that μ_i takes, and $\mu_{i,max}$ is the highest value that μ_i takes. Based on this state-space model, as in Ref.[42], a digraph with nodes consisting of the state variables $x_1, \cdots, x_{n_x}$, input variables $u_1, \cdots, u_{n_u}$, and outputs $y_1, \cdots, y_{n_y}$ can be developed. The connection between each pair of variables is classified into three types [43]: input to state variable links (u_i, x_j), state to state variable links (x_i, x_j), and state to output variable links (x_i, y_j). Note that there are no links between input variables (u_i, u_j), between output variables (y_i, y_j), or between input and output variables (u_i, y_j).

We quantify the strength of the interaction between each pair of variables by conducting sensitivity analyses based on the state-space model of Eq. (11). The sensitivity of one variable to another variable is obtained by taking the partial derivative of the first variable with respect to the second variable and evaluating the partial derivative at the desired steady-state (ss) conditions $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{n_x}, \bar{u}_1, \bar{u}_2, \dots, \bar{u}_{n_u})$:

$$S_{ij} = \frac{\partial g_j}{\partial u_i}\bigg|_{ss}, \quad \bar{S}_{ij} = \frac{\partial g_j}{\partial x_i}\bigg|_{ss}, \quad \bar{\bar{S}}_{ij} = \frac{\partial h_j}{\partial x_i}\bigg|_{ss} \tag{12}$$

where S_{ij} is a measure of the sensitivity of x_j to u_i, $\bar{S}_{ij}$ a measure of the sensitivity of x_j to x_i, and $\bar{\bar{S}}_{ij}$ a measure of the sensitivity of y_j to x_i.

When a delay-free mathematical model in the following form is available:

$$\begin{cases} \dot{x}(t) = G(x(t), u(t)) \\ y(t) = H(x(t)) \end{cases}$$

where all variables (x, u, y, t) are normalized,

$$S_{ij} = T\frac{\partial G_j}{\partial u_i}\bigg|_{ss}, \quad \bar{S}_{ij} = T\frac{\partial G_j}{\partial x_i}\bigg|_{ss}, \quad \bar{\bar{S}}_{ij} = \frac{\partial H_j}{\partial x_i}\bigg|_{ss}$$

where T is the time-discretization step size (sampling period), which is assumed to be sufficiently small. As an example, the derivation of $\bar{S}_{ij}$ is as follows:

$$\int_{x_{ss,j}}^{x_{ss,j}+\Delta x_j} dx_j = \Delta x_j = \int_{kT}^{(k+1)T} G_j(x_{ss} + \overline{\Delta x}_i, u_{ss})dt$$

where $\overline{\Delta x}_i$ is of a vector of dimension n_x whose elements are all zero except for the *i*th element Δx_j. When T is sufficiently small, using the first-order truncated Taylor series expansion of $G_j(x, u)$ around (x_{ss}, u_{ss}):

$$\int_{kT}^{(k+1)T} G_j(x_{ss} + \overline{\Delta x}_i, u_{ss})dt \approx G_j(x_{ss}, u_{ss})T + \int_{kT}^{(k+1)T} \frac{\partial G_j}{\partial x_i}\bigg|_{ss} \Delta x_i dt =$$

$$\frac{\partial G_j}{\partial x_i}\bigg|_{ss} \Delta x_i \int_{kT}^{(k+1)T} dt = \Delta x_i \frac{\partial G_j}{\partial x_i}\bigg|_{ss} T$$

Thus,

$$\Delta x_j = \Delta x_i \frac{\partial G_j}{\partial x_i}\bigg|_{ss} T, \qquad \bar{S}_{ij} = \frac{\Delta x_j}{\Delta x_i} = T\frac{\partial G_j}{\partial x_i}\bigg|_{ss}$$

The results of the sensitivity analysis are presented in the form of the matrix $[S_{ij}]$:

$$
[S_{ij}] = \begin{array}{c} \\ u_1 \\ \vdots \\ u_{n_u} \\ x_1 \\ \vdots \\ x_{n_x} \\ y_1 \\ \vdots \\ y_{n_y} \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} u_1 & \dots & u_{n_u} & x_1 & \dots & x_{n_x} & y_1 & \dots & y_{n_y} \end{array} \\
\left[\begin{array}{ccccccccc}
0 & \dots & 0 & S_{11} & \dots & S_{1n_x} & 0 & \dots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \dots & 0 & S_{n_u 1} & \dots & S_{n_u n_x} & 0 & \dots & 0 \\
0 & \dots & 0 & \bar{S}_{11} & \dots & \bar{S}_{1n_x} & \bar{\bar{S}}_{11} & \dots & \bar{\bar{S}}_{1n_y} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \dots & 0 & \bar{S}_{n_x 1} & \dots & \bar{S}_{n_x n_x} & \bar{\bar{S}}_{n_x 1} & \dots & \bar{\bar{S}}_{n_x n_y} \\
0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0
\end{array}\right]
\end{array}
$$

In this work, we use the relation:

$$
w_{ij} = \begin{cases} \left|(\log|S_{ij}|)\right|, & S_{ij} \neq 0 \\ 0, & S_{ij} = 0 \end{cases}, \qquad i,j = 1, \cdots, (n_x + n_u + n_y)
$$

to assigning a weight to each link. The logarithm is taken to scale the range of the weight, as in Ref. [44]. The second absolute value is taken to make sure all assigned weights are positive. $S_{ij} = 0$ implies that there is no link between the two variables and therefore $w_{ij} = 0$.

3.3. Multi-Objective Whale Optimization Algorithm

Several multi-objective optimization algorithms have been used in community detection. They include metaheuristic algorithms such as GA [45, 46], ant colony algorithm [29], and PSO [47, 48]. In this paper, we detect communities of the large-scale networks by using a WOA based on the nondominated sorting framework. The WOA has been found to show better performance in terms of convergence speed and accuracy than algorithms such as PSO, GA, and DE [40, 49, 50]. Moreover, the WOA has fewer tunable parameters than other metaheuristic algorithms.

In order for the WAO to handle the multi-objective optimization problems, WAO should be extended as its original form is for single-objective optimization problems. In this work, we apply the fast non-dominated sorting method [27] to generate the list of non-dominated solutions. To expand and implement the fast non-dominated sorting method, two important features, including the crowding distance and the non-dominated sorting mechanism, should be incorporated into the

WOA. In fast non-dominated sorting, a comparison between each solution with every single solution is made to check if it is dominated or not. This comparison is made for all individuals to find the members of the first rank. This procedure is repeated to find other non-dominated fronts (rank two or more). Solutions with lower non-dominated ranks are preferred over others.

Apart from convergence to the Pareto-optimal set, the solutions should be diverse (should spread along the Pareto optimal front). To ensure the diversity, the crowding distance mechanism reported in [27] is used. Crowding distance is generally used when we cannot decide the priority between the solutions in the same front. In this situation, the solution with a higher crowding distance should be preferred as these solutions maintain diversity among generated nondominated solutions. To calculate the crowding distance, first ϑ solutions in the Pareto front are sorted in descending order in terms of their objective functions, $f_1(z), \cdots, f_p(z)$. Let the order be: $z_1^i, \cdots, z_\vartheta^i$ for $f_i(z)$. The two solutions that yield the smallest and largest values of an objective function $f_i(z)$ are denoted by z_{min}^i and z_{max}^i; that is, $z_{min}^i = z_\vartheta^i$ and $z_{max}^i = z_1^i$. The crowding distances of $f_1(z), \cdots, f_p(z)$ are then calculated using:

$$CD_{qi} = \frac{f_i(z_{q+1}^i) - f_i(z_{q-1}^i)}{f_i(z_{max}^i) - f_i(z_{min}^i)} \qquad q = 2, \cdots, (\vartheta - 1) \tag{13}$$

The overall crowding distance is the sum of all individual distances:

$$CD_q = \sum_{i=1}^{p} CD_{qi} \tag{14}$$

3.3.1. Initialization Phase

Our proposed multi-objective whale optimization algorithm requires an initial solution to start. We generate the initial solution by using the locus-based adjacency representation (LAR) with some modifications [51]. In LAR, the position of each whale has a dimension of n_z, represented by n_z genes $\{G_1, \cdots, G_{n_z}\}$, where n_z indicates the number of nodes. As an example, consider the network shown in Figure 1(a). Each gene i is connected randomly to one of its neighbors and takes a value between 1 to n_z (table in Figure 1). Since the random initialization mechanism in LAR may produce unrealistic links that do not exist in the real network, we modify LAR by

applying the ordered neighbor list [52]. This modification limits the pool of neighbors that can be connected randomly to each gene i, to those that are really connected to gene i (Table 1). Based on this random realistic initial arrangement, the modified LAR determines communities, as shown in Figure 1(c).

Table 1: Order neighbor list for the example shown in Figure 1(a).

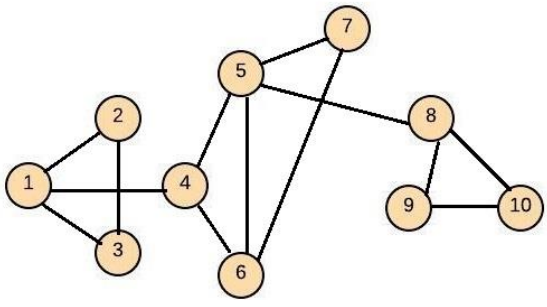
Node No.	Order neighbor list			
1	2	3	4	
2	1	3		
3	1	2		
4	1	5	6	
5	4	6	7	8
6	4	5	7	
7	5	6		
8	5	9	10	
9	8	10		
10	8	9		

3.3.2. Updating Population Position

The multi-objective optimization problems in community detection require optimization solvers that can handle variables that take integer numbers. Existing binary WOA versions [53, 54] are not suitable for this class of optimization problems. To address this limitation, we propose and use the following approach to discretize and update the position of each whale:

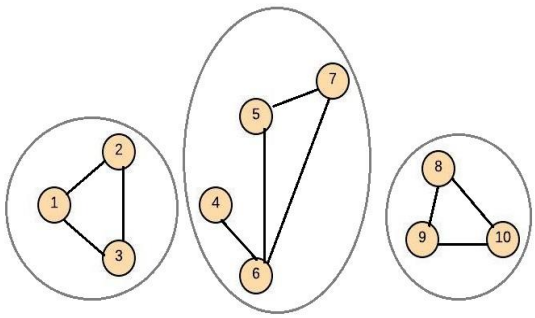
$$T\left(z_i^j(t)\right) = \left|\frac{1-\exp\left(-z_i^j(t)\right)}{1+\exp\left(-z_i^j(t)\right)}\right|, \quad i = 1, \cdots, n_z \tag{15}$$

$$\ddot{z}_i^j(t+1) = \begin{cases} \alpha\,, & \text{if } \lambda < T\left(z_i^j(t)\right) \text{ and } k_i > 1 \\ \ddot{z}_i^j(t), & \text{otherwise} \end{cases} \tag{16}$$



Position	1	2	3	4	5	6	7	8	9	10
Genotype	3	1	2	6	6	5	5	9	10	8

(a)



(b)

Figure 1: (a) An actual network. (b) The neighbor initially connected to each gene by LAR. (c) The community structure found by modified LAR.

where z^j is the position of whale j in the continuous space, and $\ddot{z}^j$ is a positive integer, which indicates the position of whale j in the discrete space. λ is a predefined threshold varying between 0 and 1, which allows nodes to replace their current connection (neighbor) with a new one if there is any neighbor around them. In this work, it is assumed to be 0.5. α is one of the nodes connected to node i, which is chosen by random from the generated ordered neighbor list.

3.3.3. Algorithm Main Loop

When a graph G representing a network N is created, the proposed algorithm starts from the initialization population described in Section 3.3.1. It then begins to optimize the two objective functions of Eq.(10) simultaneously to identify non-dominated solutions according to the non-dominated sorting algorithm [27]. The position of each whale is compared with the positions of the other whales, and the non-dominated solutions (positions) are kept and ranked in an archive. Since we select a specific number of non-dominated solutions in the Pareto optimal front, these solutions should be selected such that they are diverse (spread along the front). To ensure that the solutions are diverse, a crowding distance is assigned to each solution. As a solution located in the less crowded region is preferred, a solution that has a higher crowding density is chosen. In every iteration, the position of each whale is updated using WOA equations. Solutions are calculated based on the updated positions, and the archive is updated accordingly. When the archive is updated, the best solution (leader) is chosen by random from the first Pareto front. It is worth pointing out that the leader is randomly chosen from the first Pareto front because there is no single best solution. This operation continues until a stopping criterion is satisfied. The results of this algorithm are solutions from the first Pareto front in the non-dominated list. Each of these generated solutions corresponds to the communities with different cluster numbers. To choose the best solutions among all of these non-dominated solutions, we apply the modularity concept [16] and evaluate the quality of generated clusters. As stated before, a larger value of modularity (Q_w) indicates the better partitioning. Algorithm 1 is the pseudocode of the proposed non-dominated sorting multi-objective whale optimization algorithm for community detection problem.

Algorithm 1: Multi-objective Whale Optimization Algorithm for Community Detection

Input: Adjacency matrix of the network
Output: Solution in the first Pareto front
Begin:

Set iteration number $t_{Current} = 0$;
Generate the ordered neighbor list
Initialize N populations based on modified LAR
Compute the objective functions using Eq (10) and choose the leader
Store the nondominated vectors into an archive
Sort individuals according to non-domination rank as described in Ref. [27]
Compute the crowding distance for each non-dominated solution stored in the archive

While $t_{Current} < t_{Max\ Iter}$,

for each whale do

Update coefficients a, D^j, A^j, P^j and l
If 1 ($P^j < 0.5$) then
If 2 ($|A^j| < 1$) then
Update the position of the current whale by Eq. (7a)
Else if 2 ($|A^j| \geq 1$)
Choose random whale, and update the position of the current whale by Eq. (8)
End if 2
Else if 1 ($P^j \geq 0.5$)
Update the position of the current whale by Eq. (7.b)
End if 1
Squash solution using Eq. (15) and update $\ddot{z}_i^j(t+1)$ from Eq. (16)
End for
Calculate the whale cost function
Sort whales according to non-domination rank
Compute the crowding distance for each non-dominated solution stored in the archive
Update archive
Update leader by random from Front 1

$t_{Current} = t_{Current} + 1$
End while
Return to archive

Until the maximum number of iterations is reached

3.4. Parallel Algorithm

Most existing metaheuristic optimization methods suffer slow convergence when dealing with large-scale problems. Besides, the efficiency of metaheuristic algorithms deteriorates due to the dimensionality of the problem. The performance of metaheuristic optimization algorithms strongly depends on the iteration number and population number. Allocating the higher number to these parameters may increase the accuracy of results obtained but requires a longer time to reach those results. Fortunately, due to the inherent parallel nature of these metaheuristic algorithms, the performance of these algorithms could be improved by exploiting this property.

Since the community detection problems are NP-hard [55] and their resolutions are CPU time-consuming, we implement parallel computing to execute the algorithm.

The parallelization strategies applied to population-based metaheuristics algorithms can be classified into four models, named the master-slave model, coarse-grained model, fine-grained model, and hybrid model [56-58]. In this paper, the algorithm is parallelized using the master-slave (MS) model due to its easy implementation and programming. The MS model has a master processor and a set of slave processors, as shown in Figure 2. Generally, in metaheuristic algorithms, the master processor is responsible for performing global search operations, while objective function evaluations are done by slave processors in parallel. In WOA, the master processor is responsible for the main unit of the algorithm which storing and updating all whales' position to obtain new individuals, while slaves execute concurrent fitness evaluation. All tasks require intensive communication between the slave processors and the master processor. This parallel execution of WOA profoundly lowers the CPU time need to solve the multi-objective optimization problems.

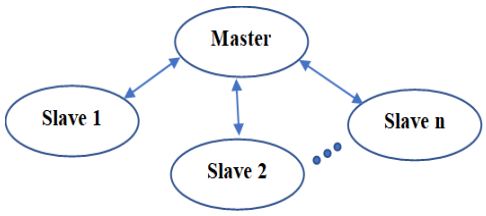


Figure 2: MS model.

4. Case Study

The proposed community detection algorithm is tested on the Tennessee Eastman process (TEP) [59]. The TEP consists of five main components: a reactor, a condenser, a centrifugal compressor, a vapor-liquid separator, and a stripper. A pipe and instrumentation diagram (P&ID) of the TEP is shown in Figure 3.

4.1. Process Model

A process model of the TEP is [59, 60]:

$$\frac{dN_{i,r}}{dt} = y_{i,6}F_6 - y_{i,7}F_7 + \sum_{j=1}^{3} \nu_{ij}\, R_j \qquad i = A, B, \dots, H \tag{17}$$

$$\frac{dN_{i,s}}{dt} = y_{i,7}F_7 - y_{i,8}(F_8 + F_9) - x_{i,10}F_{10} \qquad i = A, B, \dots, H \tag{18}$$

$$\frac{dN_{i,m}}{dt} = z_{i,1}F_1 + z_{i,2}F_2 + z_{i,3}F_3 + y_{i,5}F_5 + y_{i,8}F_8 + F_i^* - y_{i,6}F_6 \qquad i = A, B, \dots, H \tag{19}$$

$$\frac{dN_{i,p}}{dt} = (1 - \varphi_i)x_{i,10}F_{10} - x_{i,11}F_{11} \qquad i = G, H \tag{20}$$

Here, the states variables, manipulated variables, and output variables are:

$$x^T = [N_{A,r}, N_{B,r}, \dots, N_{H,r}, N_{A,s}, N_{B,s}, \dots, N_{H,s}, N_{A,m}, N_{B,m}, \dots, N_{H,m}, N_{G,p}, N_{H,p}\ ] \quad (21)$$

$$u^T = [F_1, F_2, F_3, F_4, F_8, F_9, F_{10}, F_{11}, T_{cr}, T_{cs}\ ] \quad (22)$$

$$y^T = [y_{A,6}, y_{B,6}, \dots, y_{F,6}, y_{A,9}, y_{B,9}, \dots, y_{H,9}, x_{G,11}, x_{H,11}, V_{lr}, V_{ls}, P_r, P_s\ ] \quad (23)$$

where indices $r, s, m,$ and p stand for the reactor, separator, mixing zone, and stripper, respectively. N_i is the molar holdup of component i. F_j indicates the molar flow rate of stream $j, j = 1, 2, \dots, 11$ $(kmol/h)$. T_c is the temperature (℃). $y_{i,j}$ is the mole fraction of component i in vapor stream j. $x_{i,j}$ is the mole fraction of component i in liquid stream j. P and V_l are pressure and liquid volume, respectively. More details on the process model can be found in Ref. [60].

4.2. TEP Graph Modeling

The graph of the process is constructed based on the state-space model in Eqs.17-20. Figure 4 shows the resulting graph, indicating the interaction between inputs, state variables, and outputs. The graph has 58 nodes, as there are 26 state variables, 12 input variables, and 20 output variables. The adjacency matrix is then defined based on this graph. The strength of the interconnection between each variable pair is determined using sensitivity analyses, and the steady-state values reported in Ref. [60]. The results allow determining the 58×58 weighted adjacency matrix. The TEP graph has 698 links. A whale population size of 50 is used, and the maximum iteration number as a stopping criterion is set to 500. The optimization problems is solved on an Intel® Core™ i7 computer with CPU 2.6 GHz and 16 GB of memory.

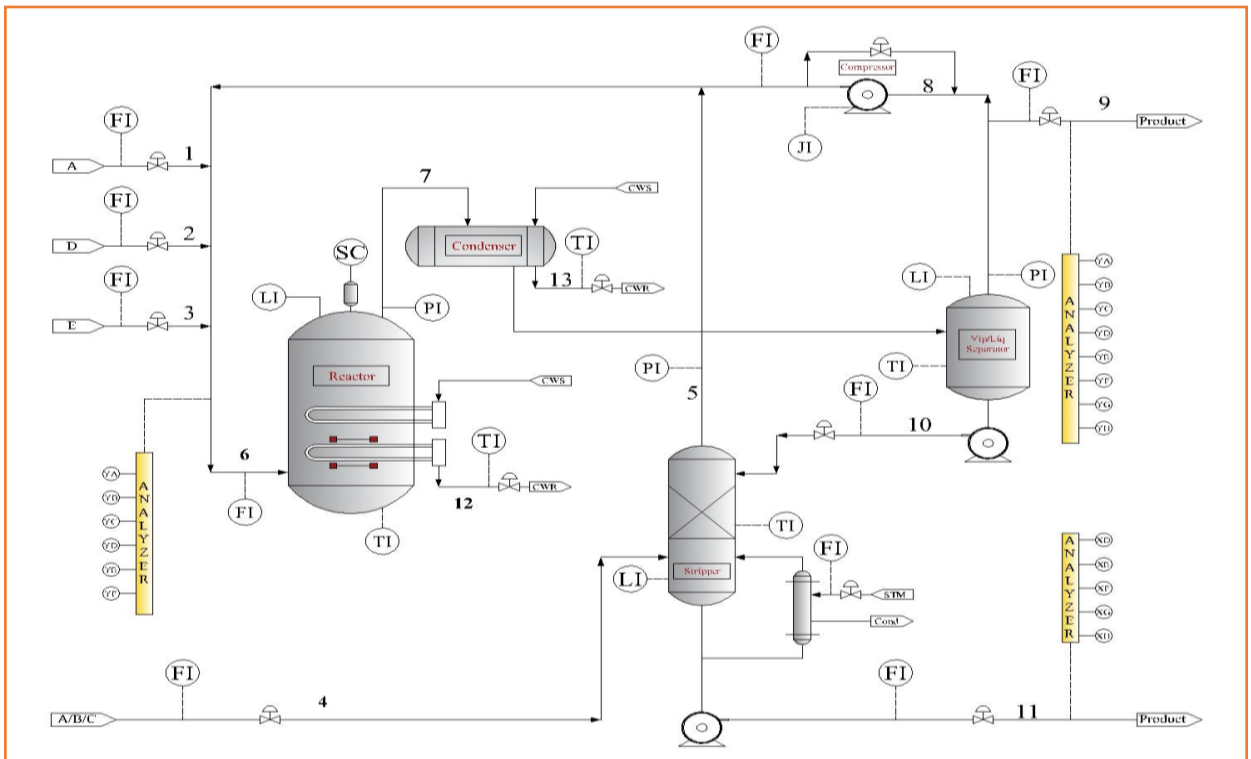


Figure 3: P&ID of the Tennessee Eastman process.

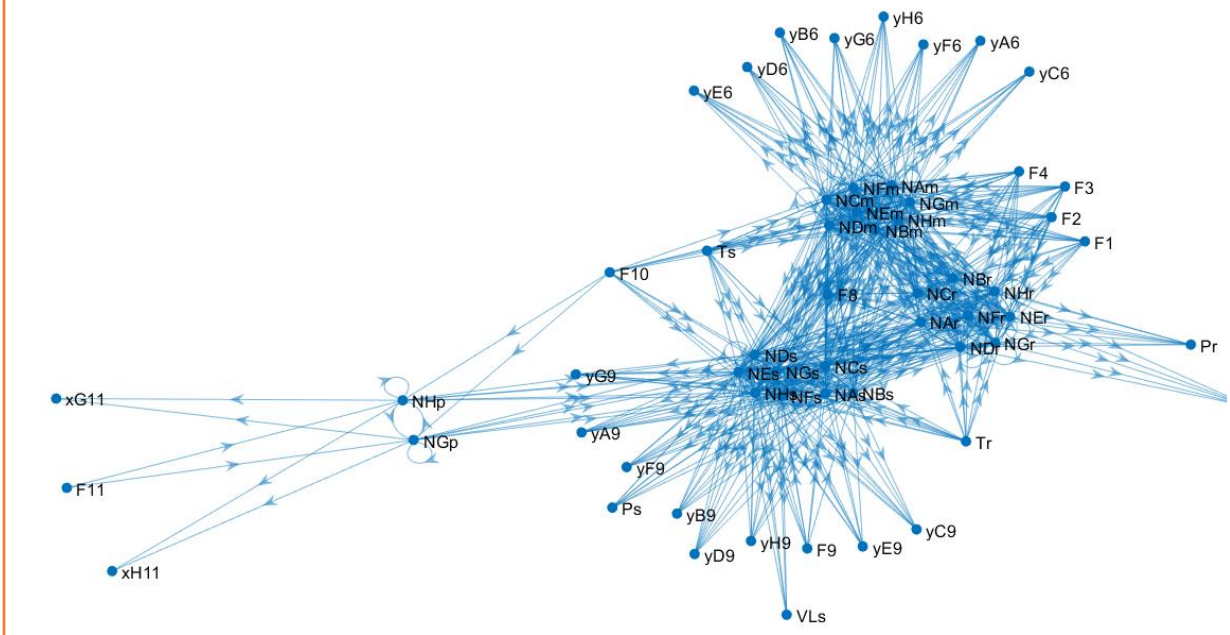


Figure 4: Graph of the Tennessee Eastman process.

4.3.Distributed Architectures

As discussed in Section 1, the use of multi-objective optimization in community detection problems allows for identifying multiple community structures. Figure 5 shows the best frontier of nondominated solutions found in the TEP using our algorithm. As the community structures corresponding to some of these non-dominated solutions are unreasonable, the modularity is used to choose reasonable partitions from the obtained set of non-dominated solutions. The negative values of the intra-connection objective function is due to the maximization of this objective, which is equivalent to the minimization of $(-Q_{Intra-connection})$. The solution A, which has the highest modularity of about 0.3747, corresponds to three communities (Figure 6); as shown in Figure 7, the reactor and feeding zone are in the first community, the separator in the second community, and the stripper in the third community. The solution B divides the TEP into two communities (Figure 8), and its modularity value is around 0.3571. In this solution, the separator is merged with the stripper, forming a larger community. However, there are other points that are close to these solutions, but they are unreasonable community structures because some communities with tight and strong connections are split into small ones. Figure 9 shows one of these unreasonable community structures with modularity 0.322, corresponding to the intra-connection 0.535 and inter-connection 0.213. As can be seen, the strong connection between the reactor and the feeding zone is split. This points to the need for applying other criteria to discriminate among the detected communities and choose the most appropriate one.

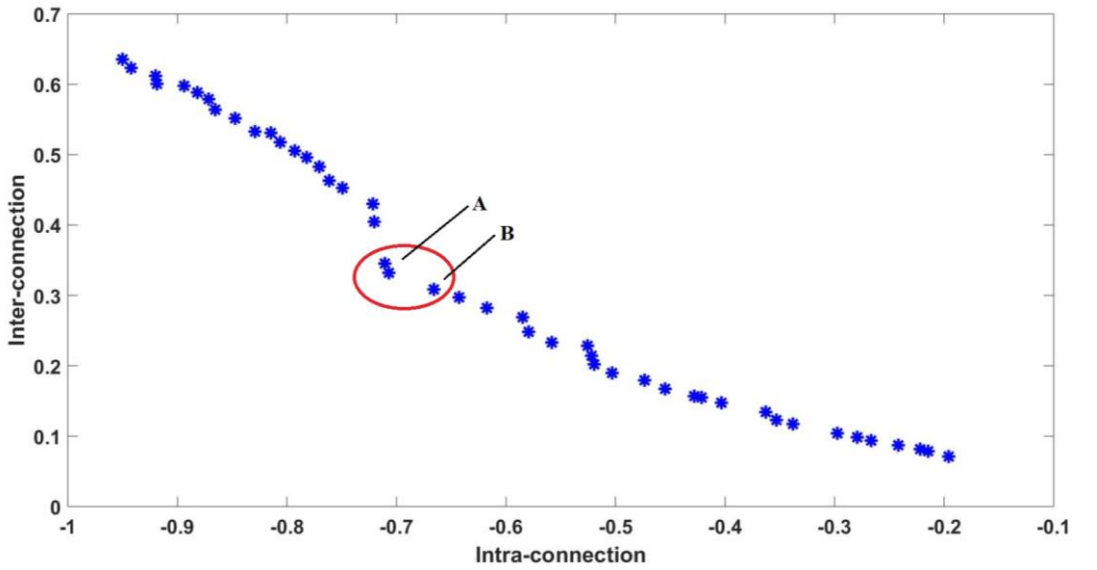


Figure 5: Pareto frontier.

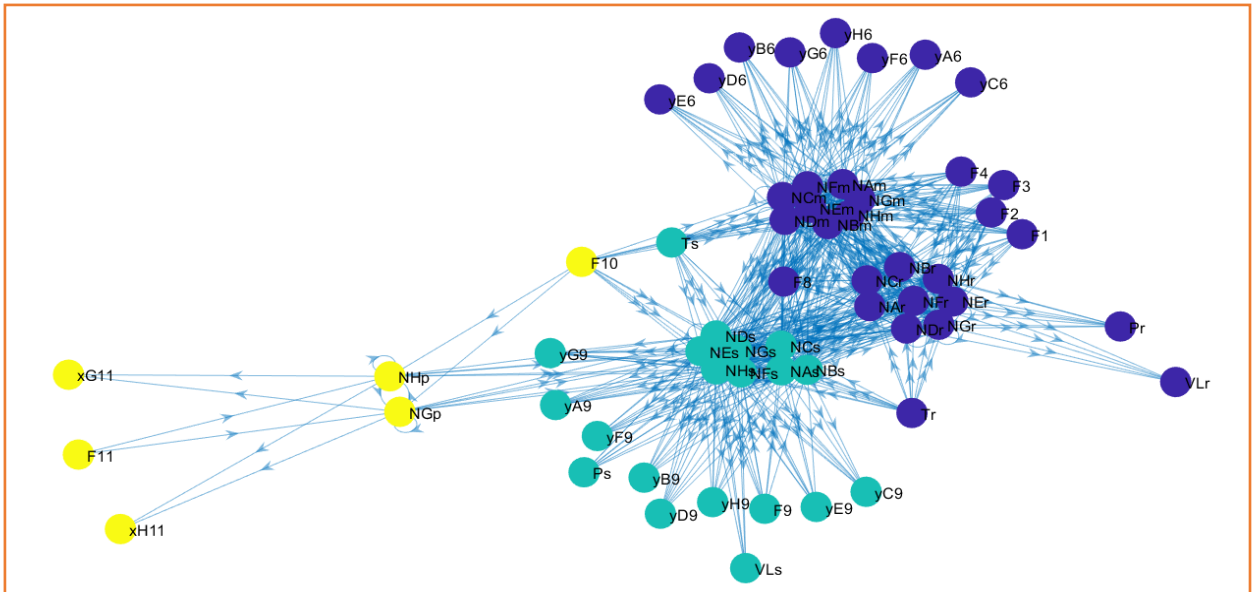


Figure 6: Three detected communities corresponding to the solution A.

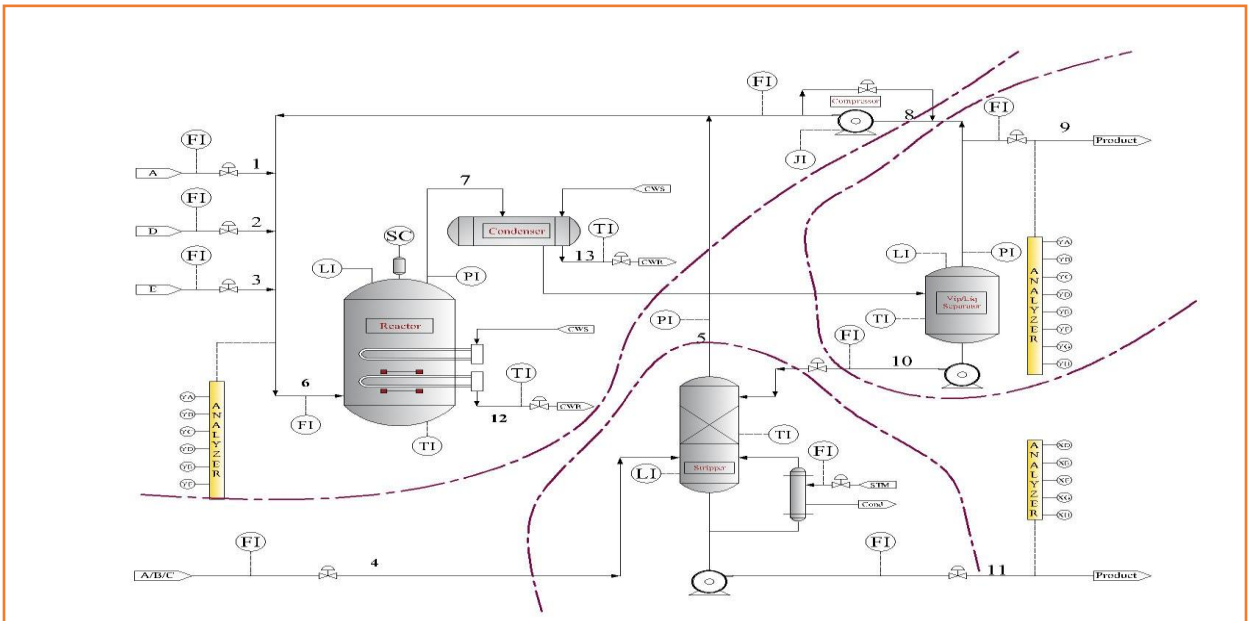


Figure 7: Detected three sub-networks in the TEP corresponding to the solution A.

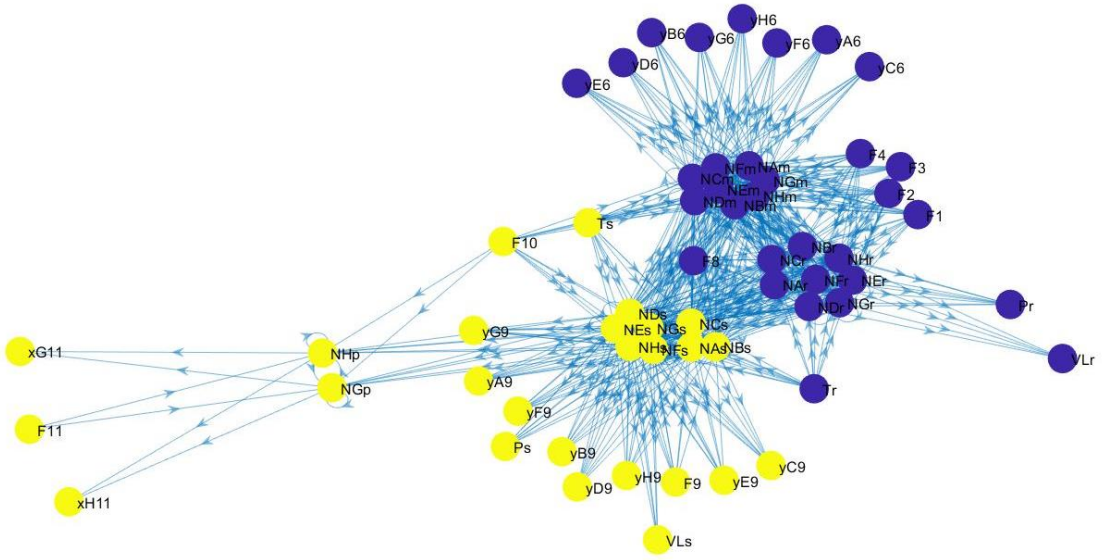


Figure 8: Two communities corresponding to the solution B.

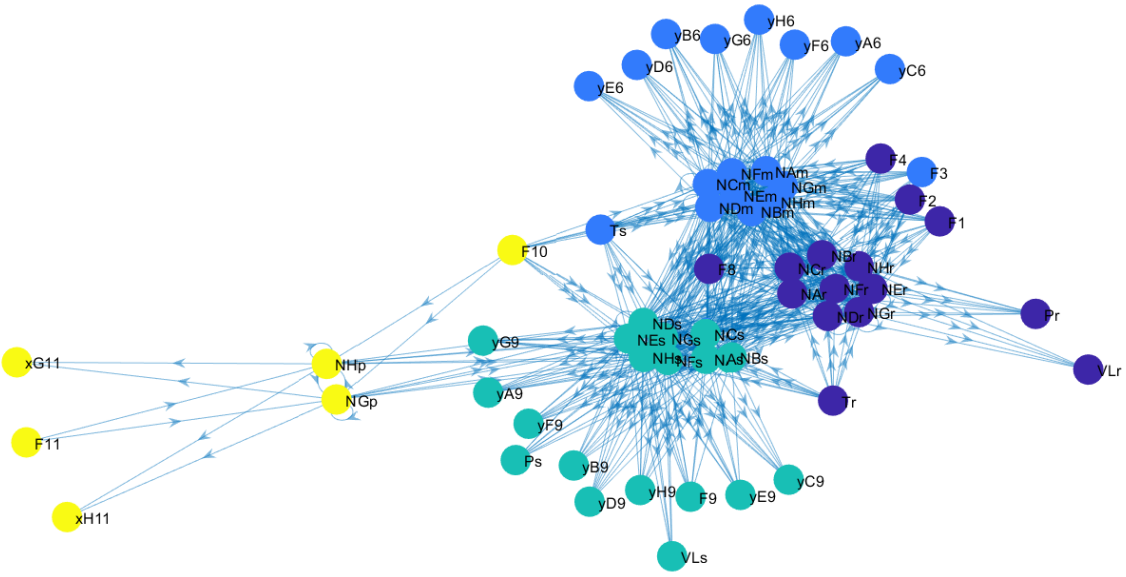


Figure 9: Unrealistic detected communities corresponding to $Q_{Intra-connection} = 0.535$ and $Q_{Inter-connection} = 0.213$.

5. Conclusion

In this work, we used the concept of community detection to decompose large-scale networks into smaller sub-networks. We proposed a novel algorithmic framework that uses a whale optimization algorithm to optimize two conflicting parts of the modularity function simultaneously. The unique search mechanism of WOA was found to have good global search ability, satisfactory convergence, and the ability to avoid local optima traps. We made the standard WOA applicable to community detection problems by developing a discrete version of the standard WOA. The discrete WOA includes the random realistic arrangement in the initialization phase and the random positive integer replacement in the updating phase. It also adopte the non-dominated sorting approach to solve community detection as a multi-objective optimization problem. The use of the non-dominated sorting concept results in efficient convergence to the true Pareto front and achieving solution diversity. We also addressed the slow convergence rate of community detection problems in large-scale networks by using a computer with parallel processors and applying the master-slave model. The algorithm was applied to the Tennessee Eastman process in which the corresponding weighted graph was constructed based on a state-space process model and sensitivity analysis at steady-state conditions. The case study showed that the proposed algorithm is capable of efficiently finding multiple sub-network configurations.

Acknowledgment

This material is based upon work supported by the U.S. National Science Foundation under Grant Nos. CBET–1704915 and CBET–1704833. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Girvan, M.; Newman, M. E., Community structure in social and biological networks. *Proceedings of the national academy of sciences* **2002,** *99* (12), 7821-7826.
2. Quirós-Tortós, J.; Terzija, V. In *A graph theory based new approach for power system restoration*, 2013 IEEE Grenoble Conference, IEEE: 2013; pp 1-6.
3. Baldea, M.; Daoutidis, P., *Dynamics and nonlinear control of integrated process systems*. Cambridge University Press: 2012.

4. Pourkargar, D. B.; Moharir, M.; Almansoori, A.; Daoutidis, P., Distributed estimation and nonlinear model predictive control using community detection. *Industrial & Engineering Chemistry Research* **2019,** *58* (30), 13495-13507.
5. Tang, W.; Allman, A.; Pourkargar, D. B.; Daoutidis, P., Optimal decomposition for distributed optimization in nonlinear model predictive control through community detection. *Computers & Chemical Engineering* **2018,** *111*, 43-54.
6. Moharir, M.; Pourkargar, D. B.; Almansoori, A.; Daoutidis, P., Distributed model predictive control of an amine gas sweetening plant. *Industrial & Engineering Chemistry Research* **2018,** *57* (39), 13103-13115.
7. Shirinivas, S.; Vetrivel, S.; Elango, N., Applications of graph theory in computer science an overview. *International journal of engineering science and technology* **2010,** *2* (9), 4610-4621.
8. Pellegrini, M.; Haynor, D.; Johnson, J. M., Protein interaction networks. *Expert review of proteomics* **2004,** *1* (2), 239-249.
9. Jeong, H.; Tombor, B.; Albert, R.; Oltvai, Z. N.; Barabási, A.-L., The large-scale organization of metabolic networks. *Nature* **2000,** *407* (6804), 651-654.
10. Soroush, M.; Masooleh, L. S.; Seider, W. D.; Oktem, U.; Arbogast, J. E., Model-predictive safety optimal actions to detect and handle process operation hazards. *AIChE Journal* **2020**, e16932.
11. Heo, S.; Daoutidis, P., Control-relevant decomposition of process networks via optimization-based hierarchical clustering. *AIChE Journal* **2016,** *62* (9), 3177-3188.
12. Cai, Q.; Ma, L.; Gong, M.; Tian, D., A survey on network community detection based on evolutionary computation. *International Journal of Bio-Inspired Computation* **2016,** *8* (2), 84-98.
13. Tang, W.; Babaei Pourkargar, D.; Daoutidis, P., Relative time-averaged gain array (RTAGA) for distributed control-oriented network decomposition. *AIChE Journal* **2018,** *64* (5), 1682-1690.
14. Daoutidis, P.; Tang, W.; Allman, A., Decomposition of control and optimization problems by network structure: Concepts, methods, and inspirations from biology. *AIChE Journal* **2019,** *65* (10), e16708.
15. Javed, M. A.; Younis, M. S.; Latif, S.; Qadir, J.; Baig, A., Community detection in networks: A multidisciplinary review. *Journal of Network and Computer Applications* **2018,** *108*, 87-111.
16. Newman, M. E.; Girvan, M., Finding and evaluating community structure in networks. *Physical review E* **2004,** *69* (2), 026113.
17. Brandes, U.; Delling, D.; Gaertler, M.; Gorke, R.; Hoefer, M.; Nikoloski, Z.; Wagner, D., On modularity clustering. *IEEE transactions on knowledge and data engineering* **2007,** *20* (2), 172-188.
18. Cai, Q.; Gong, M.; Shen, B.; Ma, L.; Jiao, L., Discrete particle swarm optimization for identifying community structures in signed social networks. *Neural Networks* **2014,** *58*, 4-13.
19. Hassan, E. A.; Hafez, A. I.; Hassanien, A. E.; Fahmy, A. A. In *A discrete bat algorithm for the community detection problem*, International Conference on Hybrid Artificial Intelligence Systems, Springer: 2015; pp 188-199.
20. Jia, G.; Cai, Z.; Musolesi, M.; Wang, Y.; Tennant, D. A.; Weber, R. J.; Heath, J. K.; He, S. In *Community detection in social and biological networks using differential evolution*, International Conference on Learning and Intelligent Optimization, Springer: 2012; pp 71-85.

21. Qu, J., A hybrid algorithm for community detection using PSO and EO. *Advances in Information Sciences and Service Sciences* **2013,** *5* (7), 187.
22. Tasgin, M.; Herdagdelen, A.; Bingol, H., Community detection in complex networks using genetic algorithms. *arXiv preprint arXiv:0711.0491* **2007**.
23. Hafez, A. I.; Zawbaa, H. M.; Hassanien, A. E.; Fahmy, A. A. In *Networks community detection using artificial bee colony swarm optimization*, Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014, Springer: 2014; pp 229-239.
24. Atay, Y.; Koc, I.; Babaoglu, I.; Kodaz, H., Community detection from biological and social networks: A comparative analysis of metaheuristic algorithms. *Applied Soft Computing* **2017,** *50*, 194-211.
25. Hassanien, A. E.; Babers, R., Metaheuristic Algorithms for Detect Communities in Social Networks: A Comparative Analysis Study. *International Journal of Rough Sets and Data Analysis (IJRSDA)* **2018,** *5* (2), 25-45.
26. Gong, M.; Chen, X.; Ma, L.; Zhang, Q.; Jiao, L., Identification of multi-resolution network structures with multi-objective immune algorithm. *Applied Soft Computing* **2013,** *13* (4), 1705-1717.
27. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T., A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* **2002,** *6* (2), 182-197.
28. Kim, M.; Hiroyasu, T.; Miki, M.; Watanabe, S. In *SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2*, International Conference on Parallel Problem Solving from Nature, Springer: 2004; pp 742-751.
29. Ji, P.; Zhang, S.; Zhou, Z., A decomposition-based ant colony optimization algorithm for the multi-objective community detection. *Journal of Ambient Intelligence and Humanized Computing* **2020,** *11* (1), 173-188.
30. Gong, M.; Ma, L.; Zhang, Q.; Jiao, L., Community detection in networks by using multiobjective evolutionary algorithm with decomposition. *Physica A: Statistical Mechanics and its Applications* **2012,** *391* (15), 4050-4060.
31. Shim, V. A.; Tan, K. C.; Cheong, C. Y., A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **2012,** *42* (5), 682-691.
32. Fan, Y.; Li, M.; Zhang, P.; Wu, J.; Di, Z., The effect of weight on community structure of networks. *Physica A: Statistical Mechanics and its Applications* **2007,** *378* (2), 583-590.
33. Kamelian, S.; Salahshoor, K., A novel graph-based partitioning algorithm for large-scale dynamical systems. *International Journal of Systems Science* **2015,** *46* (2), 227-245.
34. Song, Y.; Li, J.; Zhang, X.; Liu, C. In *Community detection using parallel genetic algorithms*, 2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI), IEEE: 2012; pp 374-378.
35. Staudt, C. L.; Meyerhenke, H., Engineering parallel algorithms for community detection in massive networks. *IEEE Transactions on Parallel and Distributed Systems* **2015,** *27* (1), 171-184.
36. Arenas, A.; Duch, J.; Fernández, A.; Gómez, S., Size reduction of complex networks preserving modularity. *New Journal of Physics* **2007,** *9* (6), 176.

37. Radicchi, F.; Castellano, C.; Cecconi, F.; Loreto, V.; Parisi, D., Defining and identifying communities in networks. *Proceedings of the national academy of sciences* **2004,** *101* (9), 2658-2663.
38. Gong, M.; Jiao, L.; Du, H.; Bo, L., Multiobjective immune algorithm with nondominated neighbor-based selection. *Evolutionary computation* **2008,** *16* (2), 225-255.
39. Manne, J. R., Multiobjective Optimization in Water and Environmental Systems Management-MODE Approach. In *Handbook of Research on Advanced Computational Techniques for Simulation-Based Engineering*, IGI Global: 2016; pp 120-136.
40. Mirjalili, S.; Lewis, A., The whale optimization algorithm. *Advances in engineering software* **2016,** *95*, 51-67.
41. Watkins, W. A.; Schevill, W. E., Aerial observation of feeding behavior in four baleen whales: Eubalaena glacialis, Balaenoptera borealis, Megaptera novaeangliae, and Balaenoptera physalus. *Journal of Mammalogy* **1979,** *60* (1), 155-163.
42. Daoutidis, P.; Kravaris, C., Structural evaluation of control configurations for multivariable nonlinear processes. *Chemical engineering science* **1992,** *47* (5), 1091-1107.
43. Siljak, D., On pure structure of dynamic systems. In *Nonlinear Systems and Applications*, Elsevier: 1977; pp 299-300.
44. Tang, W.; Daoutidis, P., Network decomposition for distributed control through community detection in input–output bipartite graphs. *Journal of Process Control* **2018,** *64*, 7-14.
45. Pizzuti, C., A multiobjective genetic algorithm to find communities in complex networks. *IEEE Transactions on Evolutionary Computation* **2011,** *16* (3), 418-430.
46. Pizzuti, C. In *Ga-net: A genetic algorithm for community detection in social networks*, International conference on parallel problem solving from nature, Springer: 2008; pp 1081-1090.
47. Li, L.; Jiao, L.; Zhao, J.; Shang, R.; Gong, M., Quantum-behaved discrete multi-objective particle swarm optimization for complex network clustering. *Pattern Recognition* **2017,** *63*, 1-14.
48. Gong, M.; Cai, Q.; Chen, X.; Ma, L., Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. *IEEE Transactions on Evolutionary Computation* **2013,** *18* (1), 82-97.
49. Kumar, C.; Rao, R. S., A novel global MPP tracking of photovoltaic system based on whale optimization algorithm. *International Journal of Renewable Energy Development* **2016,** *5* (3).
50. Kaveh, A.; Ghazaan, M. I., Enhanced whale optimization algorithm for sizing optimization of skeletal structures. *Mechanics Based Design of Structures and Machines* **2017,** *45* (3), 345-362.
51. Handl, J.; Knowles, J., An evolutionary approach to multiobjective clustering. *IEEE transactions on Evolutionary Computation* **2007,** *11* (1), 56-76.
52. Huang, F.; Li, X.; Zhang, S.; Zhang, J.; Chen, J.; Zhai, Z., Overlapping community detection for multimedia social networks. *IEEE Transactions on multimedia* **2017,** *19* (8), 1881-1893.
53. Hussien, A. G.; Hassanien, A. E.; Houssein, E. H.; Amin, M.; Azar, A. T., New binary whale optimization algorithm for discrete optimization problems. *Engineering Optimization* **2019**, 1-15.

54. Reddy K, S.; Panwar, L.; Panigrahi, B.; Kumar, R., Binary whale optimization algorithm: a new metaheuristic approach for profit-based unit commitment problems in competitive electricity markets. *Engineering Optimization* **2019,** *51* (3), 369-389.
55. Karrer, B.; Levina, E.; Newman, M. E., Robustness of community structure in networks. *Physical review E* **2008,** *77* (4), 046119.
56. Cantú-Paz, E., A summary of research on parallel genetic algorithms. **1995**.
57. Alba, E., *Parallel metaheuristics: a new class of algorithms*. John Wiley & Sons: 2005; Vol. 47.
58. Cantu-Paz, E., *Efficient and accurate parallel genetic algorithms*. Springer Science & Business Media: 2000; Vol. 1.
59. Downs, J. J.; Vogel, E. F., A plant-wide industrial process control problem. *Computers & chemical engineering* **1993,** *17* (3), 245-255.
60. Ricker, N.; Lee, J., Nonlinear modeling and state estimation for the Tennessee Eastman challenge process. *Computers & chemical engineering* **1995,** *19* (9), 983-1005.