

Article

Research on Modeling Method for Optimal Allocation of Wellhead Targets in Large Well Clusters

Liupeng Wang ^{1,*}, Haonan Duan ¹ , Zhikun Liu ¹, Yuanchao Peng ² and Xuyang Liu ¹

¹ College of Petroleum Engineering, Xi'an Shiyou University, Xi'an 710065, China; 13571409242@163.com (H.D.); lzk12431@xsyu.edu.cn (Z.L.); lxy87205@163.com (X.L.)

² Chuanqing Drilling Engineering & Technology Research Institute, Xi'an 710021, China; pyc_gcy@cnpcc.com.cn

* Correspondence: wangliupeng@xsyu.edu.cn; Tel.: +86-158-2991-3600

Abstract: The paper proposes a genetic ant colony algorithm that integrates genetic and ant colony algorithms, enhancing the heuristic function of the latter, to address target point distribution issues in large well clusters. This algorithm utilizes genetic algorithms for initial pheromone distribution and employs the ant colony algorithm to achieve rapid convergence. Introducing genetic operators in each iteration addresses the ant colony system's drawbacks, including scarcity of initial pheromones, susceptibility to local optima, and slow convergence speed. The model aims to minimize the sum of horizontal displacement and intersections in line connections from wellheads to target points as its dual-objective function. It validates the effectiveness of the genetic ACO algorithm in optimizing target point allocation at wellheads through a case study, highlighting its advantages over traditional methods in reducing displacement, ensuring result stability, and preventing collisions.

Keywords: large well clump; wellhead target assignment; genetic ant colony algorithm; horizontal displacement minimization; bi-objective function



Citation: Wang, L.; Duan, H.; Liu, Z.; Peng, Y.; Liu, X. Research on Modeling Method for Optimal Allocation of Wellhead Targets in Large Well Clusters. *Processes* **2024**, *12*, 1705. <https://doi.org/10.3390/pr12081705>

Academic Editor: Olympia Roeva

Received: 11 July 2024

Revised: 11 August 2024

Accepted: 12 August 2024

Published: 14 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A large well cluster refers to a group of two or more wells drilled on a single rig under a specific wellhead spacing [1]. Due to the close proximity of wellheads, drilling a wellbore may occasionally intersect with other wellbores drilled nearby. To mitigate the risk of collisions, the optimal allocation of wellheads to proposed wells is crucial in both conventional and unconventional oil and gas operations [2–5], aiming to minimize inter-well interference and construction complexity. Researchers globally often focus on principles, such as minimizing horizontal displacement, total well depth, or their combination for wellhead allocation, developing corresponding mathematical models and solution approaches. In 2012, Shi Yucai et al. proposed a mathematical model for allocating wellheads in large clusters, focusing on minimizing horizontal displacements from wellheads to target drilling points as the objective function, and improving the genetic algorithm for solution [6]. In 2014, Yan Tie et al. extended this by incorporating the minimization of horizontal projections between wellheads and target points using an ant colony algorithm with constraints on non-intersecting projections [7]. In 2016, Gu Shutian utilized genetic optimization algorithms to devise a two-step design method for regional platform comprehensive planning and specific wellhead layout within the platform [8]. In 2021, D.S. Yue et al. applied the minimum horizontal displacement sum as the optimization criterion, employing an enhanced Hungarian algorithm for large-scale well cluster allocation [9]. In 2022, Liu Zhikun et al. developed a large well cluster wellhead–target allocation model that minimized the horizontal projection plane characteristic distance from the wellhead to the target stage, utilizing the Hungarian algorithm to optimize the model [10].

The genetic algorithm (GA) is an optimization technique that emulates biological evolution principles from Darwin's theory [11–14]. It aims to find optimal solutions by

simulating natural selection and genetic mechanisms. Besides typical intelligent optimization traits, GA offers strong global search capabilities, maintains solution diversity, and minimizes the risk of local optima. However, its lack of feedback mechanisms limits its ability to utilize historical search data, reducing search efficiency and making optimization outcomes probabilistic. Ant colony optimization (ACO) is a biomimetic algorithm inspired by ants' foraging behavior, widely used for solving optimization problems [15–18]. ACO leverages a positive feedback mechanism to enhance optimization efficiency through accumulated search experience. While versatile in combination with other methods, ACO's drawbacks include extended search times and susceptibility to local optima.

Existing models often neglect collision avoidance and typically focus solely on minimizing horizontal displacement from wellheads to target points. Conversely, single solution algorithms for the wellhead–target distribution problem suffer from low search efficiency, susceptibility to local optima, and unstable outcomes. This paper adopts the wellhead-to-target horizontal displacement and the number of intersections with the wellhead-to-target line as dual objective functions. It optimizes the pheromone distribution of an enhanced ant colony algorithm using a genetic algorithm integration to address initial pheromone scarcity in allocating wellhead target points.

2. Genetic Algorithm and Ant Colony Algorithm

2.1. Genetic Algorithm

The genetic algorithm (GA) [19–21] is an optimization method inspired by biological evolution in nature, simulating the continuous evolution of populations through natural selection and genetic adaptation to environmental changes. Introduced by Holland of the University of Michigan in 1975, GA is an iterative process that begins with a random initial population. It operates on the principle of survival of the fittest, employing competition, selection, reproduction, mutation, and other genetic optimizations to enhance the performance of successive generations, aligning with environmental constraints or specific application guidelines.

The basic genetic algorithm [22] initially encodes parameters to generate an initial population, employing a fitness function to evaluate individual performance. Individuals demonstrating superior performance are probabilistically selected to undergo crossover and mutation processes, serving as parents to produce the next generation of the population.

The basic genetic algorithm follows these steps:

- (1) Initialize parameters, such as population size N , number of iterations n , crossover probability P_c , mutation probability P_r , and others.
- (2) Initialize the position information of population individuals.
- (3) Evaluate each individual's fitness using a designated fitness function, identifying optimal positions and corresponding solutions.
- (4) Select parent individuals based on predefined selection criteria.
- (5) Perform crossover and mutation operations on selected parents to generate offspring. Then, select these parents and offspring as the final individuals.
- (6) Check termination conditions; if not met, return to step (3); otherwise, proceed to (7).
- (7) Output the optimal individual position and solution set within the population.
- (8) Conclude the algorithm.

2.2. Ant Colony Algorithm

The ant colony algorithm (ACO) [23], a simulated evolutionary algorithm, was introduced by Italian scholars M. Dorigo et al. based on their study of real ant colony behavior. They discovered that individual ants communicate using a substance called pheromone, which they deposit on paths as they move. Ants detect pheromones to guide their movements, leading to collective behaviors with positive feedback: paths with more pheromone attract more ants. This exchange enables ants to efficiently search for food. ACO simulates this process to solve combinatorial optimization problems such as the Traveling Salesman

Problem (TSP) [24], Assignment Problem [25], and Job Shop Scheduling Problem [26], achieving promising experimental results. The general process involves the following:

- (1) Establish an ant colony system mode, where n nodes are traversed by m ants to find the shortest path connecting all nodes exactly once.
- (2) Define the transfer probability for ant k moving from current node i to next node j as follows:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}^k(t)]^\beta}{\sum_{s \in a_k} [\tau_{is}(t)]^\alpha [\eta_{is}^k(t)]^\beta}, & j \in a_k \\ 0, & \text{others} \end{cases} \quad (1)$$

where α is the information heuristic factor, β is the expectation heuristic factor, and $\eta_{ij}^k(t)$ is the visibility.

N is the number of loop traversals, which means that all ants complete one traversal loop (i.e., complete the traversal of each node), and the number of loop traversals is denoted as t ($t \geq 0, t \in \mathbb{Z}$); τ_{ij} is the amount of pheromone on the path between node i and node j ;

- (3) Pheromone localization update: The amount of pheromone is updated once for each completed loop traversal using Equation (2), and the calculation formula is expressed as follows:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2)$$

where ρ is the pheromone volatilization coefficient, a constant less than 1; $\Delta\tau_{ij}(t)$ denotes the pheromone increment of all the ants in the path between node i and node j in this loop traversal, which can be calculated using Equation (3):

$$\tau_{ij}(t) = \sum_{k=0}^m \Delta\tau_{ij}^k(t) = \sum_{k=0}^m \eta_{ij}^k(t) \quad (3)$$

The $\Delta\tau_{ij}^k(t)$ is calculated using the ant-cycle model, in which

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, & \text{Ant } k \text{ chooses the path in this loop}(i, j) \\ 0, & \text{Ant } k \text{ did not choose a path in this loop}(i, j) \end{cases} \quad (4)$$

where Q is a constant; L_k is the total length of the path taken by ant k .

- (4) Pheromone global update. All path pheromones are updated, and the calculation equation is expressed as follows:

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (5)$$

- (5) All endpoints are traveled or there is no shorter path, the algorithm ends.

Ants randomly select their walking direction during the search for the optimal path. To narrow the range of the ant's forward direction, the objective is to visit all wellheads and target points once. This paper introduces an enhanced heuristic function that directs the ant's movement at each new position, aiming to accelerate the convergence speed.

$$\eta_{ij}(t) = \begin{cases} d_{ij}, & \text{wellhead to target direction} \\ d_{ij}^\infty, & \text{Target to wellhead direction} \\ d_{ij}^{-\infty}, & \text{Wellhead to wellhead direction} \\ d_{ij}^{-\infty}, & \text{Target to target direction} \end{cases} \quad (6)$$

where $\eta_{ij}(t)$ is the heuristic function and d_{ij} is the distance from the wellhead to the target point.

2.3. Comparison of the Two Algorithms

By analyzing the principles and steps of genetic algorithms and ant colony optimization, two bio-inspired algorithms, we find commonalities and differences in their optimization processes.

- (1) Probabilistic global optimization: Both algorithms employ probabilistic selection mechanisms such as crossover probability, mutation probability, roulette wheel selection, and transfer probability, enhancing their ability to find global optimal solutions.
- (2) Non-dependence on strict mathematical properties: Both algorithms do not rely solely on strict mathematical properties or precise mathematical descriptions of objective functions and constraints during optimization.
- (3) Intrinsic parallelism: Both algorithms exhibit inherent parallelism, making them suitable for large-scale parallel computation, thereby achieving significant efficiency gains with minimal computational resources.
- (4) Robustness: Both algorithms demonstrate robustness and effectiveness across diverse conditions and environments.

Advantages of the genetic algorithm: Genetic algorithms explore solution spaces from multiple starting points and support large population sizes. They use probabilistic selection based on individual fitness, offering flexibility in exploration. Crossover operators promote diversity. However, genetic algorithms may struggle with “hill climbing” and premature convergence due to low mutation rates, potentially trapping in local optima. They require large populations, limiting efficiency for super-large-scale problems compared to other methods.

Advantages of ACO algorithm: ACO is a stochastic search method inspired by collective behaviors in biological communities. It converges to optimal solutions through pheromone accumulation, particularly effective for combinatorial optimization problems. Its strong positive feedback accelerates later-stage evolution, facilitating rapid convergence. However, ACO may suffer from slow initial convergence due to initial pheromone scarcity, leading to prolonged search times and susceptibility to local optima.

2.4. Genetic Ant Colony Algorithm Design

The ant colony algorithm’s initial pheromone deficiency is addressed by optimizing it with a genetic algorithm, enhancing algorithm accuracy. This involves using the superior solutions from the genetic algorithm to compute the initial pheromone needed by the ant colony algorithm, which subsequently determines the optimal path. The genetic ant colony algorithm process and the associated pseudo-code are as follows:

- (1) Initial population generation and genetic algorithm operations: The genetic algorithm initiates by generating an initial population, with each individual representing a path assignment from the wellhead to the target point;
`population = generate_initial_population(population_size)`
`for generation in range(num_generations);`
- (2) Fitness evaluation: Each individual’s fitness is evaluated based on the total path distance, with higher fitness assigned to shorter distances;
`fitness_scores = evaluate_fitness(population);`
- (3) Main loop of genetic algorithm: In this loop, parent individuals are selected via roulette wheel selection for crossover and mutation operations to produce offspring;
`new_population = []`
`for _ in range(population_size)`
- (4) Ant colony local optimization: At regular intervals (e.g., every 10 generations), the algorithm initiates local optimization using the ant colony method
`if generation % 10 0:`
`global_best_path = global_best_solution(population)`
`pheromone_matrix=local_pheromone_update(pheromone_matrix,global_best_path);`

- (5) Population update: Following local optimization by the ant colony, the entire population is updated based on the globally optimal paths identified;
- (6) Convergence curve and optimal path output: Throughout each genetic algorithm iteration, the convergence is tracked by plotting a convergence curve and identifying the shortest path for output;
 $\text{best_solution} = \text{global_best_solution}(\text{population})$
 $\text{print}(\text{"optimal path assignment results:"}, \text{best_solution}).$

Thus, integrating the global search capability of the genetic algorithm with the local optimization prowess of the ant colony resolves the wellhead-to-target path optimization challenge, enhancing both efficiency and quality of the algorithm's search.

3. Multi-Objective Modeling of Wellhead Target Allocation Optimization

3.1. Plane Upper Line Segment Equation Establishment

Equation of line l_i segment:

$$\begin{cases} x = x_i + t(x_j - x_i) \\ y = y_i + t(y_j - y_i) \end{cases}, 0 \leq t \leq 1 \quad (7)$$

When $t = 0$, the equation represents the starting point of the line (x_i, y_i) , and when $t = 1$, the equation represents the ending point of the line (x_j, y_j) . For any value of $0 < t < 1$, the equation represents the point on the line from (x_i, y_i) to (x_j, y_j) .

3.2. Intersection of Two Line Segments in the Plane Solved

3.2.1. Determination of Parallelism of Plane Segments

Suppose two line segments AB is l_i and CD is l_j and their endpoints are $A(x_i, y_i)$, $B(x_{i+1}, y_{i+1})$, $C(x_j, y_j)$ and $D(x_{j+1}, y_{j+1})$.

$$\begin{cases} k_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \\ k_j = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} \end{cases} \quad (8)$$

If $k_i = k_j$, it shows that line segment l_i is parallel to l_j if $k_i \neq k_j$, it shows that line segment l_i is not parallel to l_j .

3.2.2. Intersection of Plane Segments Solution

Translate four points A, B, C and D to x_i in x -axis direction and y_i in y -axis direction, let the translated points be $A'(x'_i, y'_i)$, $B'(x'_{i+1}, y'_{i+1})$, $C'(x'_j, y'_j)$ and $D'(x'_{j+1}, y'_{j+1})$, using the chi-square matrix representation as follows:

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -x_i \\ 0 & 1 & -y_i \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

At this point:

$$\begin{cases} x'_i = x_i - x_i = 0 \\ y'_i = y_i - y_i = 0 \end{cases} \quad (9)$$

Similarly, we can obtain the B' , C' and D' coordinates.

(1) Intersection of non-parallel line segments solution

The line segment $A'B'$ is l'_i :

$$x = tx'_{i+1}, y = ty'_{i+1}, 0 \leq t \leq 1 \quad (10)$$

The line segment $C'D'$ is l_j' :

$$x = x_j' + \mu(x_{j+1}' - x_j'), y = y_j' + \mu(y_{j+1}' - y_j'), 0 \leq \mu \leq 1 \quad (11)$$

At this point the coordinates of the intersection point are as follows:

$$\begin{cases} t = \frac{x_j'(y_{j+1}' - y_j') - y_j'(x_{j+1}' - x_j')}{x_{i+1}'(y_{j+1}' - y_j') - y_{i+1}'(x_{j+1}' - x_j')} \\ \mu = \frac{y_j'x_{i+1}' - x_j'y_{i+1}'}{y_{i+1}'(x_{j+1}' - x_j') - x_{i+1}'(y_{j+1}' - y_j')} \end{cases} \quad (12)$$

To solve this system of two equations, equate both sides of the Equation's expression for x and y , solving for t and μ . If the computed t and μ are both in the interval $[0, 1]$, then the point of intersection is on both segments; otherwise, the segments do not intersect.

(2) Intersection of parallel (co-linear) straight line segments solution

a. Parallel straight-line segments co-equal determination

Calculate the distance between the endpoint of a straight-line segment and another straight-line segment by taking the point C' and the line segment $A'B'$:

$$d = \frac{|y_{i+1}'x_j' - x_{i+1}'y_j'|}{\sqrt{y_{i+1}'^2 + x_{i+1}'^2}} \quad (13)$$

If $d \neq 0$, it means that l_i' and l_j' are parallel but not coextensive; if $d = 0$, it means that l_i' and l_j' are coextensive.

b. Co-linear segment overlap judgment

If l_i' and l_j' are coincident, there are two cases: coincident overlap (including end-point overlap and segment overlap) and coincident disjunction. In this case, you need to determine whether a point is on the line segment, you can use the following formula.

$$\begin{cases} \min(x_i', x_{i+1}') \leq x_j' \leq \max(x_i', x_{i+1}') \\ \min(y_i', y_{i+1}') \leq y_j' \leq \max(y_i', y_{i+1}') \end{cases} \quad (14)$$

If $x_j' = x_i'$ or $y_j' = y_i'$ or $y_j' = y_{i+1}'$, then the endpoints of l_i' and l_j' coincide, and there is only one intersection point;

If $\min(x_i', x_{i+1}') < x_j' < \max(x_i', x_{i+1}')$ and $\min(y_i', y_{i+1}') < y_j' < \max(y_i', y_{i+1}')$, then the line segment l_i' coincides with l_j' , and there are infinitely many intersections;

If $x_j' > \max(x_i', x_{i+1}')$ or $x_j' < \min(x_i', x_{i+1}')$ or $y_j' > \max(y_i', y_{i+1}')$, there is no intersection of the line l_i' and l_j' .

3.3. Model Assumptions

- (1) Wellhead location $W_i(X_{Wi}, Y_{Wi}), i = 1, \dots, N_w$; where N_w is the number of wellheads to be assigned, X_{wi} is the horizontal coordinate corresponding to wellheads, and Y_{wi} is the vertical coordinate corresponding to wellheads.
- (2) Target location $T_j(X_{Tj}, Y_{Tj}), j = 1, \dots, N_T$; where N_T is the number of wellheads to be drilled, X_{Tj} is the horizontal coordinate corresponding to the wellhead, and Y_{Tj} is the vertical coordinate corresponding to the wellhead.
- (3) Requirements for the allocation of wellheads and targets: the number of wellheads to be allocated is the same as the number of wells to be drilled, and only one target can be allocated to one wellhead.

- (4) The projections of the lines connecting the wellheads and targets on the horizontal plane do not intersect or intersect as little as possible.

3.4. Establishment of Multi-Objective Optimization Model for Well Target Allocation

In general, the number of wellheads to be assigned is equal to the number of wells to be drilled ($N_W = N_T = N$). Take the target point horizontal displacement as the objective function, the minimum sum of target point horizontal displacement as the optimization index, the projection of the line segment connecting the target point of the wellhead in the horizontal plane does not intersect or intersects as little as possible as the optimization index and establish the optimization model of wellhead allocation. The objective function matrix $\{G_{ij}\}$ ($N \times N$ matrix) can be derived in advance from the known conditions. G_{ij} is the horizontal displacement of the target point corresponding to assigning the i th wellhead to the j th well to be drilled, which is calculated as follows:

$$G_{ij} = \sqrt{(x_{Tj} - x_{Wi})^2 + (y_{Tj} - y_{Wi})^2}, i, j = 1, \dots, N. \quad (15)$$

The optimization problem of wellhead allocation for large well clusters is equivalently transformed into solving the matrix of decision variables $\{\alpha_{ij}\}$, given the matrix of objective functions $\{G_{ij}\}$. The wellhead allocation scheme can be represented by the decision variable matrix $\{\alpha_{ij}\}$ ($N \times N$ matrix). α_{ij} is an integer from 0 to 1; if the i th wellhead is assigned to the j th well to be drilled, then $\alpha_{ij} = 1$; otherwise, $\alpha_{ij} = 0$.

The bi-objective function is as follows:

$$\begin{cases} \min F_1 = \sum_{i=1}^N \sum_{j=1}^N \alpha_{ij} G_{ij} \\ \min F_2 = \sum_{i=1}^N \sum_{j=1}^N g(l_i, l_j) \end{cases} \quad (16)$$

$g(l_i, l_j)$ is the function of counting the number of intersections between l_i and l_j . l_i and l_j are the lines connecting the wellhead and the target point, when the lines l_i and l_j intersect.

The constraints are as follows:

$$\begin{cases} \sum_{i=1}^N \alpha_{ij} = 1, i, j = 1, \dots, N; \\ \sum_{j=1}^N \alpha_{ij} = 1, i, j = 1, \dots, N. \end{cases} \quad (17)$$

In the above planning model, the role of constraints is to ensure that each wellhead can only be assigned to a particular well to be drilled. The optimization problem of wellhead allocation for large clusters of wells is a typical assignment problem.

4. Example Solving

To validate the performance of the genetic ant colony algorithm proposed in this paper for solving the wellhead target allocation problem, the case of target allocation was taken at the wellhead of a large well cluster in a certain oilfield as an example. The algorithm is compared with the ant colony algorithm and genetic algorithm through a program run 20 times, each with 300 iterations, in Table 1, involving 30 wells and 30 target points.

Table 1. Data table of wellhead and target coordinates.

Serial Number	Wellhead Coordinates (m)	Target Point Coordinates (m)
1	(520.3, −918.8)	(67.6, 311.7)
2	(−468.5, −725.6)	(32.6, 311.7)
3	(−390.8, −435.8)	(−30.4, 311.7)
4	(−370.6, −297.8)	(−65.4, 311.7)
5	(−313.2, −146.1)	(−313.2, −146.1)
6	(−325.6, −100.8)	(31.0, 305.9)
7	(−235.5, 143.7)	(−32.0, 305.9)
8	(−356.9, 347.8)	(−67.0, 305.9)
9	(−68.0, 934.9)	(64.5, 300.1)
10	(−145.7, 645.1)	(29.5, 300.1)
11	(−223.3, 355.3)	(−33.5.5, 300.1)
12	(−181.7, −204.1)	(−68.5, 300.1)
13	(−222.3, 372.5)	(63.0, 294.3)
14	(−163.1, 431.7)	(28.0, 294.3)
15	(−93.0, 477.2)	(−35.0, 294.3)
16	(−14.9, 507.2)	(−70.0, 294.3)
17	(67.7, 520.3)	(61.4, 288.5)
18	(151.2, 515.9)	(26.4, 288.5)
19	(231.9, 494.2)	(36.6, 288.5)
20	(306.5, 456.3)	(−71.6, 288.5)
21	(371.4, 403.6)	(59.9, 282.7)
22	(424.1, 338.7)	(24.9, 282.7)
23	(462.0, 264.1)	(−38.1, 282.7)
24	(488.0, 99.8)	(−73.1, 282.7)
25	(−532.9, 758.2)	(58.4, 276.9)
26	(310.3, 985.6)	(23.4, 276.9)
27	(901.6, 524.7)	(−39.6, 276.9)
28	(91.7, −1135.3)	(−74.6, 276.9)
29	(376.5, −72.8)	(56.8, 271.1)
30	(298.8, −362.6)	(21.8, 271.1)

Figure 1 presents the results of 20 independent experiments of the program. It is observed that the ant colony algorithm exhibits the largest fluctuation in finding the optimal path each time, followed by the genetic algorithm, and then the algorithm proposed in this paper, which consistently finds optimal solutions smaller than those of the other two algorithms in the majority of cases. This suggests good stability of the hybrid algorithm.

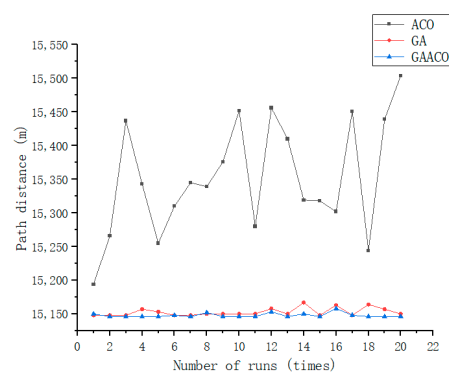
**Figure 1.** Comparison of three algorithms running process.

Figure 2 depicts the convergence iteration graph. It illustrates that the ant colony algorithm exhibits rapid initial convergence, but results in the longest path at final convergence. Both the genetic algorithm and the algorithm proposed in this paper initially have identical path lengths for the first 20 iterations. However, variations in path length divergence occur between 20 and 220 iterations, with minimal differences observed thereafter. Ultimately,

the algorithm proposed in this paper achieves superior final path length and convergence metrics compared to the genetic algorithm.

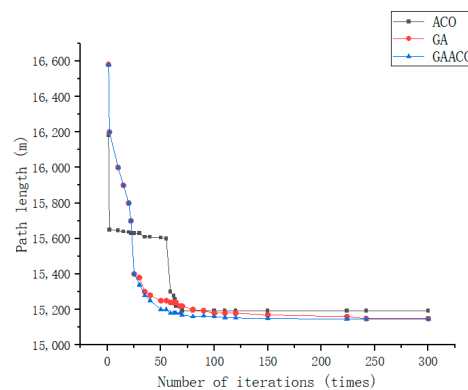


Figure 2. Comparison of convergence iteration curves of three algorithms.

Table 2 shows, In the same number of well targets, the ant colony algorithm has the largest shortest single-well horizontal displacement sum, the smallest number of convergence iterations, the largest number of intersection points, and the second largest running time; the genetic algorithm is the second largest and has the shortest running time; the algorithm in this paper has a smaller shortest single-well horizontal displacement sum and a smaller number of convergence iterations than the genetic algorithm, and there is no number of intersection points either, but it has the longest running time. Table 3 shows the one-to-one allocation results of 30 wellhead and 30 target points by the GAACO algorithm in this paper.

Table 2. Comparison results of the three algorithms.

Arithmetic	Number of Well Targets	Minimum Single Well Horizontal Displacement and (m)	Number of Convergence Iterations	Number of Intersections	Runtime (s)
ACO	10	6669.28	9	0	4.7
GA	10	6669.28	22	0	3.13
GAACO	10	6669.28	12	0	5.86
ACO	20	8948.20	24	3	17.57
GA	20	8908.12	75	1	4.81
GAACO	20	8902.70	69	0	49.19
ACO	30	15,194.00	60	9	18.56
GA	30	15,149.41	242	1	6.89
GAACO	30	15,146.31	224	0	235.36

Table 3. Assignment of GAACO wellhead target results.

Wellheads	Target	Distance (m)	Wellheads	Target	Distance (m)
1	26	1313.51	16	10	211.81
2	27	1090.40	17	2	211.53
3	23	800.40	18	14	253.54
4	19	674.76	19	1	245.56
5	15	520.91	20	5	283.66
6	28	453.50	21	9	323.88
7	24	213.76	22	17	366.16
8	20	291.40	23	21	402.53
9	3	624.33	24	25	464.67
10	7	357.55	25	8	649.34
11	16	164.99	26	6	734.85
12	30	516.94	27	13	869.67
13	12	169.99	28	22	1419.57
14	4	154.74	29	29	469.55
15	11	186.83	30	18	705.49

Figure 3 illustrates that the algorithm proposed in this study adheres strictly to the fundamental principles of wellhead allocation: ensuring each wellhead corresponds to exactly one drilled well and that each target is allocated uniquely to a wellhead without intersecting the projections of connecting lines on the horizontal plane. Thus, the dual-objective function of minimizing target horizontal displacement and intersections of connecting lines serves as the optimization criterion for effectively satisfying the allocation principles in large well clusters.

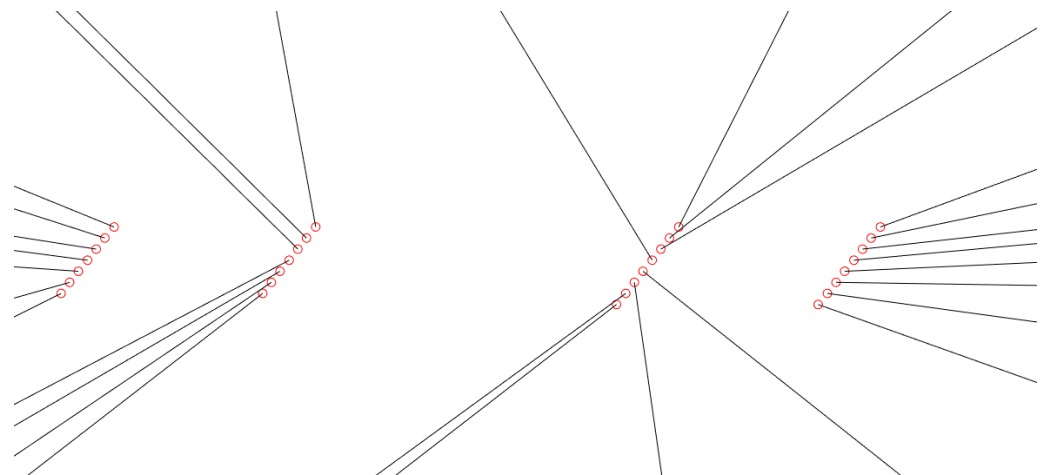


Figure 3. Distribution of GAACO wellhead target results (localized zoom).

5. Conclusions

- (1) In addressing the wellhead target allocation problem, this paper's genetic ant colony algorithm effectively minimizes target horizontal displacements and reduces crossings among target points at wellheads in large well clusters. This improves the safety and efficiency of well cluster development processes compared to traditional genetic and ant colony algorithms.
- (2) The genetic ant colony algorithm proposed in this paper reduces computational time in solving large-scale wellhead target allocation problems. To further enhance the model's efficiency, parallel computing can be explored to optimize speed and resource utilization.

Author Contributions: Methodology, L.W. and H.D.; software, L.W. and H.D.; validation, H.D.; formal analysis L.W. and H.D.; investigation, H.D.; resources, Y.P.; data curation, X.L.; writing—original draft preparation, H.D.; writing—review and editing, L.W. and Z.L.; visualization, H.D.; supervision, Z.L.; project administration, Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ni, F.; Yang, G.; Zhang, B. ROP Improvement Technologies for Large-Cluster Horizontal Shale Oil Wells in the Changqing Oilfield. *Pet. Drill. Tech.* **2021**, *49*, 29–33.
2. Jia, Z.; Zheng, M.; Zhang, F. Unconventional hydrocarbon resources in China and the prospect of exploration and development. *Pet. Explor. Dev.* **2012**, *39*, 129–136. [[CrossRef](#)]
3. Fan, L.; Wang, L.; Bu, P.; Wang, Y.; Li, J. Analysis and prospects of oil and gas resources situation at home and abroad in the first half of 2024. *China Min.* **2024**, *33*, 1–8.
4. He, B.; Jiang, X.; He, J.; Lv, Q.; Liu, L. Growth paths of traditional petroleum industry under new quality productive forces. *Nat. Gas Technol. Econ.* **2024**, *3*, 11–16+39.

5. Luo, Q.; Wang, J.; He, B.; Shang, K.; Zhang, L.; Wen, F.; Qiu, X. Non-traditional hydrocarbon resources: New physical domains with great potential for petroleum exploration and development. *Nat. Gas Explor. Dev.* **2024**, *3*, 1–11.
6. Shi, C.; Guan, C.; Zhu, L.; Chen, Y.; Zhang, X. Wellhead allocation model for large scale cluster well and its self-adaptive genetic algorithm. *J. China Univ. Pet.* **2012**, *6*, 70–73.
7. Yan, T.; Jiao, F.; Bi, L.; Xu, T.; Yu, Y.; Lu, D. Allocation method for platform wellheads of cluster well based on ACA. *Broken Block Oil Gas Field* **2014**, *2*, 224–227.
8. Gu, T. *Design and Optimization of Long Lateral Horizontal Wellpath in Long-Offset Distance*; China University of Petroleum: Beijing, China, 2016; pp. 19–36.
9. Yue, S.; Shi, C.; Chen, X.; Shao, H.; Di, C.; Zhang, S. Wellhead Assignment Mode and Improved Hungarian Algorithm for Large Scale Cluster Well Group. In Proceedings of the International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), Zhuhai, China, 24–26 September 2021; pp. 296–299.
10. Xie, X.; Kong, H.; Yang, Q.; Xu, Y. Research on Resource Scheduling of Global Adjustable non-Dominant Genetic Algorithm in Edge Computing. *Comput. Appl. Softw.* **2024**, 1–10.
11. Wang, X. A nonlinear correction method for coal mine underground sensors based on genetic improved neural network. *Ind. Instrum. Autom. Equip.* **2024**, *4*, 114–119.
12. Xu, X.; Qiu, C.; He, T.; Qin, T.; Liu, Z. Research on Multi-facility Location Problem Based on Genetic Algorithm. *Logist. Technol.* **2024**, *15*, 5–9.
13. Liang, H. Optimization of Sightseeing Paths in Tourism Park Based on Improved Ant Colony Algorithm. *Control. Inf. Technol.* **2024**, *3*, 80–85.
14. Shao, R. Robot Path Planning Based on Improved Ant Colony Algorithm. *J. Shanxi Datong Univ. Nat. Sci. Ed.* **2024**, *4*, 5–8.
15. Wang, R.; Long, L. Research on reconstruction of passive optical communication networks based on genetic ant colony algorithm. *Laser J.* **2024**, *7*, 210–214.
16. Liu, K.; Liu, L.; Chen, L.; He, B.; Wang, P. Design Method of Wellhead Target Allocation of Large well Cluster Based on Hungarian Algorithm. *J. Xi'an Shiyou Univ.* **2022**, *3*, 70–74.
17. Liu, P.; Yu, H. An Improved Adaptive Genetic Algorithm. *Ship Electron. Eng.* **2021**, *6*, 101–104.
18. Liu, S.; Huang, Q. Application of Multi-strategy Ant Colony Algorithm in Robot Path Planning. *Comput. Eng. Appl.* **2022**, *6*, 278–286.
19. Liu, H.; Gao, L.; Han, Y.; Zhang, D. Research on Mobile Robot Path Planning Based on Improved Genetic Algorithm. *Manuf. Autom.* **2024**, *5*, 26–30.
20. Dai, H.; Li, Q.; Kou, S. Survey on the Theory of Genetic Algorithms. *Control. Decis. Mak.* **2000**, *3*, 263–273.
21. Zhong, Z.; Zhu, L.; Fu, W.; Qin, J.; Zhao, M. Parallel Disassembly Sequence Planning Using a Discrete Whale Optimization Algorithm for Equipment Maintenance in Hydropower Station. *Processes* **2024**, *12*, 1412. [[CrossRef](#)]
22. Gao, W. An Improved Fast-Convergent Genetic Algorithm and Its Performance Study. *Syst. Eng. Electron.* **2003**, *11*, 1427–1430.
23. Xue, X.; Zhu, X.; Zhu, H. RRobot path planning based on improved ant colony optimization. *Xi'an Univ. Eng.* **2024**, *6*, 1–8.
24. Ding, L.; Chen, J.; Qiu, H. Ant colony optimization algorithm based on Levy flight transfer rule for solving traveling salesman problem. *Comput. Appl. Res.* **2024**, *41*, 1420–1427.
25. Qin, L.; Zong, Q.; Li, Y.; Zhang, Y.; Zhang, Y. Task Allocation of Multi-Robot Based on Improved Ant Colony Algorithm. *Space Control. Technol. Appl.* **2018**, *44*, 55–59.
26. Lei, W.; Lin, Q.; Yao, B. The Network Scheduling Model and Its Solution Algorithm of Internet Customized Shuttle Bus. *Transp. Syst. Eng. Inf.* **2017**, *17*, 157–163.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.