Cooperative Coevolutionary PSO Based Segment Assignment in Shield Tunneling

Koya Ihara[1,2*], **Shohei Kato**[1,2], **Hiroichi Masuda**[3] and **Yasuyuki Singu**[3]
[1]Dept. of Computer Science and Engineering, Graduate School of Engineering,
Nagoya Institute of Technology
[2]Frontier Research Institute for Information Science, Nagoya Institute of Technology
[3]Shimizu Corporation
{ihara, shohey}@katolab.nitech.ac.jp

Abstract

The application of artificial intelligence could reduce labor costs and improve the productivity of shield tunneling, a widely used tunnel construction method. In the shield tunneling planning process, tunnel segments are assigned along a predetermined curve, called the planning line. Conventionally, skilled engineers manually assign segments to minimize gaps between each segment and the planning line although they need to reduce each gap to only less than a tolerance. This implies construction costs can be reduced by adjusting the assignment within the tolerance. This study considers reduced gaps as constraints and attempts to reduce the amount of excavated soil by considering segment assignment as a constrained combinatorial optimization problem. Under these conditions, the segment assignment problem has severe constraints and high dimensional variables. This paper presents a co-evolutionary particle swarm optimization-based method for large-scale constrained combinatorial optimization. A two-dimensional simulation experiment using real-world construction data was performed to evaluate the effectiveness of the proposed method. The results demonstrate that the proposed method statistically outperforms the work of skilled engineers and other comparative methods in all test problems.

1 Introduction

Labor shortage is a serious problem in the construction industry worldwide. The United States' construction industry, which employs more than seven million workers, has experienced a severe shortage of skilled labor since the early 1980s, and this shortage is expected to continue [Olsen *et al.*, 2012].

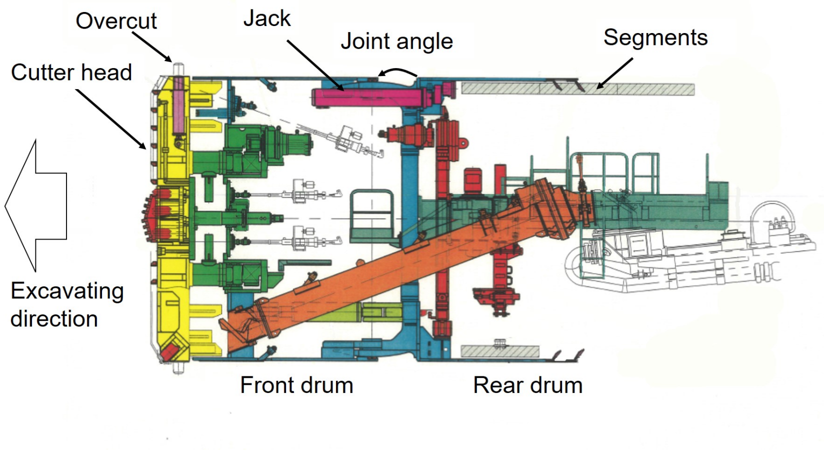
In Hong Kong, the scope and extent of public and private sector infrastructure is rapidly growing; however, currently, labor supply cannot keep pace with the demand as 12% of the construction workers in Hong Kong have reached retirement age (60 years) and another 44% are over 50 years of age, *i.e.*, close to retirement [Ng and Chan, 2015].

The Japanese construction industry is facing problems such as manpower shortages, aging workers, and reduced international competitiveness. Since November 2015, the Japanese Ministry of Land, Infrastructure, Transport and Tourism has promoted *i-Construction* [Suzuki, 2016], an initiative to optimize and upgrade the entire construction process—from investigation and design to construction and inspection, including maintenance. i-Construction’s primary concepts are utilization of information and communication technology and the introduction of innovative technology, such as artificial intelligence (AI) through cooperation between industries, governments, and academia. In this study, a practical construction support system based on i-Construction principles and focused on shield tunneling is developed.

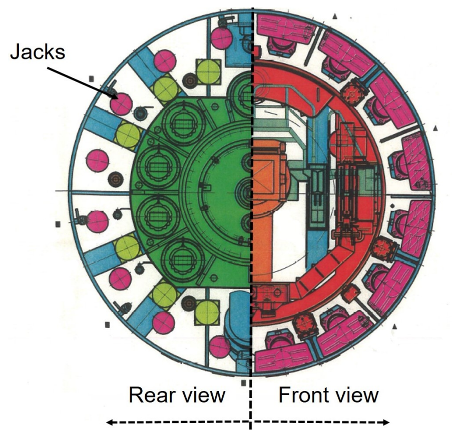
The *shield tunneling* [Maidl *et al.*, 2013; Japan Society of Civil Engineers, 2007] is a common tunnel construction method. Shield tunneling techniques have been intensively studied [Lin *et al.*, 2019; Koyama, 2003] in the civil and mechanical engineering domains. In addition, some previous studies [Chen *et al.*, 2019; Hasanipanah *et al.*, 2016] have examined shield tunneling in the AI domain. Zhou *et al.* presented a predictive framework for the attitude and position in shield tunneling by applying a hybrid deep learning model, which contains convolutional neural network feature extractor, and long short-term memory predictor [Zhou *et al.*, 2019]. Zhang *et al.* established predictive models for assessing surface settlement caused by earth pressure balance shield machine based on extreme gradient boosting, artificial neural network, support vector machine, and multivariate adaptive regression spline [Zhang *et al.*, 2020]. However, to the best of our knowledge, no studies have focused on shield tunneling planning processes and improved its efficiency.

In the planning process, tunnel segments are assigned to a predetermined planning line, and, conventionally, to minimize the gaps between segments and the planning line, skilled engineers assign segments manually. Nevertheless, we have only to reduce each gap less than a tolerance. In addition, reducing the amount of excavated soil along each segment is desirable. Automation and segment assignment optimization will address the problem of skilled labor shortage and improve productivity. In this study, we addressed segment

*Contact Author



(a) Side View



(b) Rear and Front View

Figure 1: An Example of Construction Diagrams of A Shield Machine

assignment as a constrained combinatorial optimization problem.

This problem has three notable characteristics: high-dimensional decision variables, categorical values, and narrow feasible region. This paper proposes the Cooperative Coevolutionary Integer Categorical Particle Swarm Optimization (εCCICPSO), which is a combination of the Cooperative Coevolutionary framework, Integer Categorical Particle Swarm Optimization (ICPSO), and ε constrained method, for large-scale constrained combinatorial optimization problem, such as the segment assignment. Herein, we have attempted to verify the effectiveness of the proposed εCCICPSO to segment assignment through the two-dimensional simulation experiment using real-world construction data.

2 Segment Assignment

In this section, we explain the segment assignment and its formulation as constrained combinatorial optimization problem.

2.1 Shield Tunneling

Shield tunneling is a tunnel construction method that uses excavation machines (*shield machines*) shown in Fig. 1. The front surface of the shield machine has cutters (called the *cutter head*) for ground excavation. The *over cut i.e.*, the external cutter equipped outside the front surface, is controlled such that the machine body can pass without contacting the ground wall. A shield machine is divided to *front* and *rear drums*, and the angle between the front drum and the rear drum (referred to as the *joint angle*) is controlled to allow the shield machine to move around curves. Segments are assembled at the rear of the shield machine, and the shield machine is propelled by the reaction force given from its jack pushing the located segment.

2.2 Segment Assignment Problem

In the planning process, multiple types of segments are provided for each construction project. The segments should be assigned along a planning line comprising straight lines and curves such that gaps between each segment and the planning line fall within a given tolerance, as shown in Fig. 2. Conventionally, skilled engineers manually assign segments to minimize these gaps without considering construction costs; however, this assignment roughly determines the shield machine' s excavation route. Thus, optimization of this assignment will reduce shield construction costs. Herein, we focus on segment assignment problems to reduce the amount of excavated soil. There are two primary demands in segment assignment: (1) gaps between each segment and the planning line should be within the given tolerance, and (2) the amount of soil excavated by the shield machine along the segments should be reduced. In this study, the former is treated as an inequality constraint, and the latter is treated as an objective function. We define segment assignment as the following constrained combinatorial optimization problem:

$$\begin{aligned} &\text{minimize} && f(\boldsymbol{x}), \\ &\text{subject to} && (g_i(\boldsymbol{x}) - g_t) \leq 0, \\ &&& x_i \in \{1, \cdots, k\}, \quad (i = 0, \cdots, n) \end{aligned} \tag{1}$$

where $x_i \in \{1, \cdots, k\}$ corresponds to the type of the segment assigned to i-th position, and k is the number of segment types. A decision variable vector $\boldsymbol{x} = (x_1, x_2, \cdots x_n)$ expresses the assigned segments. The objective function $f(\boldsymbol{x})$ is the amount of soil excavated along to segments $\boldsymbol{x}$ by the shield machine; $g_i(\boldsymbol{x})$ as the gap between the i-th segment and the planning line; and g_t is the gap tolerance.

This problem involves an n-dimensional decision variable vector, where n is generally over several hundreds. Consequently, this problem has an extremely large search space; however, a solution must be quickly obtained because the segment assignment plan should be revised when the actual construction deviates from the plan. Population-based metaheuristics, such as swarm intelligence (SI), are frequently used for real-world optimization problems as they provide easy parallelization and demonstrate

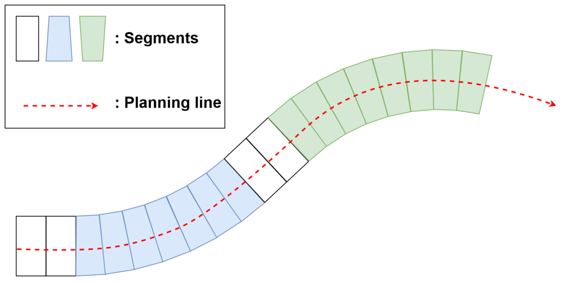


Figure 2: An Example of Segment Assignment

good multi-point search efficiency [Pitakaso *et al.*, 2020; Singh and Salgotra, 2019; Hassanien and Emary, 2018; Zhang *et al.*, 2014]. Although many discrete optimization algorithms are based on SI, they often only consider integer problems [Kennedy and Eberhart, 1995; Pampara *et al.*, 2005]. However, segment assignment involves variables whose values are categorical and unordered rather than purely numerical. PSO is one of the most widely used SI algorithms, and integer categorical PSO (ICPSO) [Strasser *et al.*, 2016], outperforms other discrete versions of PSO in unordered discrete optimization. In addition, gap tolerance g_t is typically about $50\,\mathrm{mm}$, whereas the diameter of segments is around $10\,\mathrm{m}$. Segment assignment has n severe constraints. For handling the severe constraints, the ε constrained method is proposed by [Takahama and Sakai, 2005]. Adapting the ε constrained method to the ICPSO, Ihara *et al.* developed ε constrained Integer Categorical Optimization (εICPSO), which demonstrated the effectiveness for the segment assignment [Ihara *et al.*, 2019]. However we found that its performance decreases when handling the high dimensional variables due to random sampling of ICPSO. Thus, we adapt the cooperative coevolutionary framework to ICPSO for large-scale constrained combinatorial optimization.

3 Proposed Method

This paper proposes Cooperative Coevolutionary Integer Categorical Particle Swarm Optimization (CCICPSO) and segment assignment using combination of CCICPSO and ε constrained method (εCCICPSO).

3.1 ICPSO

The ICPSO is a novel PSO algorithm that has been shown to surpass other discrete PSO algorithms [Strasser *et al.*, 2016]. In the PSO, particles search for the best position of the search space. Particles have a position and a velocity, and the position corresponds to a candidate solution. Original PSO assumes continuous state variables. In the ICPSO, the representation of the particle's position is altered so that each attribute in a particle is a distribution over its possible values rather than a value itself similarly to Estimation of Distribution Algorithms (EDAs) [Larrañaga and Lozano, 2002]. A particle is evaluated by sampling a candidate solution from these distributions and then calculating its fitness.

In the ICPSO, a particle p's position $\mathbf{X}_p$ is represented as $\mathbf{X}_p = [\mathcal{D}_{p,1}, \mathcal{D}_{p,2}, \cdots, \mathcal{D}_{p,n}]$ where each $\mathcal{D}_{p,i}$ is the probability distribution for variable X_i. In other words, each component of the position vector is a set of probabilities $\mathcal{D}_{p,i} = [d^a_{p,i}, d^b_{p,i}, \cdots, d^k_{p,i}]$, where $d^j_{p,i}$ denotes the probability that variable X_i takes on value j for particle p. A particle p's velocity $\mathbf{V}_p$ is a vector of n vector φ, which control the particle's probability distributions. $\mathbf{V}_p = [\varphi_{p,1}, \varphi_{p,2}, \cdots, \varphi_{p,n}]$, $\varphi_{p,1} = [\psi^a_{p,i}, \psi^a_{p,i}, \cdots, \psi^a_{p,n}]$, where $\psi^j_{p,i}$ corresponds to velocity of particle p for variable i in state j.

The velocity and position update equations are applied directly to the values in the distribution.

$$\begin{aligned}\mathbf{V}_p =& \omega\mathbf{V}_p + U(0,\phi_1) \otimes (\mathbf{pBest} - \mathbf{X}_p)\\ &+ U(0,\phi_2) \otimes (\mathbf{gBest} - \mathbf{X}_p),\\ \mathbf{X}_p =& \mathbf{X}_p + \mathbf{V}_p,\end{aligned}$$

where each operator is performed component-wise over each variable in the vector; and $U(0,\phi_1)$ and $U(0,\phi_2)$ are uniformly distributed random numbers between 0 and ϕ_1 and 0 and ϕ_2 respectively. The vector **pBest** is the best position in the search space this particle has ever reached; the **gBest** is the best position in the search space any particle in the swarm has ever reached. The particle moves in the search space by adding the updated velocity to the particle's position vector at the current iteration. The particle's behavior is controlled with adjusting the parameter ω, ϕ_1, and ϕ_2 known as inertia, the cognitive component and the social component.

After the velocity and position update, any value outside [0,1] is mapped to the nearest boundary in order to maintain a valid probability. In addition, the distribution is normalized to ensure that the sum of its values is 1.

To evaluate a particle p, its distributions are sampled to create a candidate solution $\mathbf{S}_p = [s_{p,1}, s_{p,2}, \cdots, s_{p,n}]$ where $s_{p,j}$ denotes the state of variable X_j. The samples are evaluated by the fitness function, and then the distributions are evaluated by their own sample's fitness value.

When a sample produced by a particle exceeds the global or local best, the best values are updated using both the distribution from the particle position P_p and the sample itself $\mathbf{S}_p$. Formally, for all states $j \in Vals(X_i)$ the global best's probability is updated as

$$d^j_{gB,i} = \begin{cases} \epsilon \times d^j_{p,i} & (j \neq s_{p,i}) \\ d^j_{p,i} + \displaystyle\sum_{\substack{k\in Vals(X_i)\\ \wedge k\neq j}} (1-\epsilon) \times d^k_{p,i} & (j = s_{p,i}) \end{cases}$$

where ϵ, the *scaling factor*, is a user-controlled parameter that determines the magnitude of the shift in the distribution restricted to $[0,1)$, and $d^j_{gB,i}$ is the global best position's probability that variable X_i takes value j. This update increases the probability of the distribution producing samples similar to the best sample, while maintaining a valid probability distribution.

3.2 CCICPSO

In the CCICPSO, a problem is decomposed into several smaller subcomponents based on Potter and De Jong's model [Potter and Jong, 2000]. Each component is assigned to a subpopulation, such that individuals in each subpopulation represent potential components to the original problem. Then each component is evolved simultaneously but independently in a round-robin fashion. The CCICPSO evolves subswarms instead of subpopulations according to the ICPSO, such that samples produced by particles in each sub-swarms represent partial solution of the entire problem.

The fitness of a subpopulation member is determined by the combination of this member and selected members from the other subpopulations. In other words, *collaborator* subcomponents are selected from each of the other subpopulations and assembled with the target individual to form a complete solution. There are many ways of selecting collaborators [Wiegand *et al.*, 2001]. This paper uses the most widely used one, choosing the best individuals from each of the individuals, that is to say, the global best samples of each of the sub-swarms.

3.3 Constraint Handling

The εCCICPSO is constrained discrete optimization algorithm based on CCICPSO and ε constrained method [Takahama and Sakai, 2005]. The ε constrained method adds the ability of constraint handling to the algorithms originally designed for unconstrained optimization problems, by introducing the ε level comparison. The εCCICPSO ranks the solution candidates with the ε level comparison instead of the general comparison.

The ε level comparison is a comparison operator considering both the constraints and objective values for ranking candidate solutions. In this method, constraint violation $\phi(\boldsymbol{x})$ is defined as a measure of how much the constraints a solution violates. The constraint violation can be given by the maximum of all constraints or the sum of all constraints.

$$\phi(\boldsymbol{x}) = \max_i \{0, g_i(\boldsymbol{x}) - g_t\},$$

$$\phi(\boldsymbol{x}) = \sum_i \max\{0, g_i(\boldsymbol{x}) - g_t\}, \tag{2}$$

where p is a positive number. In this paper, constrain violation is given by (2), the sum of all constraints.

The ε level comparison ($<_\varepsilon$,$\leq_\varepsilon$) is defined as an order relation on the set of $(f(\boldsymbol{x}), \phi(\boldsymbol{x}))$. If $f_1(f_2)$ and $\phi_1(\phi_2)$ are the objective values and the constraint violation of solution point $x_1(x_2)$ respectively, then the comparison operators $<_\varepsilon$ and $\leq_\varepsilon$ are defined by the following:

$$(f_1, \phi_1) \leq_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{cases} f_1 \leq f_2, & (\phi_1, \phi_2 \leq \varepsilon) \\ f_1 \leq f_2, & (\phi_1 = \phi_2) \\ \phi_1 < \phi_2, & \text{otherwise} \end{cases}$$

This definition means that the ε level comparison compares two solutions by constraint violation value first. If both solutions have violation value under a small threshold ε the two solutions are then compared by the objective function value only.

4 Experiment

We attempt to verify the effectiveness of εCCICPSO to segment assignment through the two-dimensional simulation experiment using real construction data with comparison with the ε constrained integer categorical particle swarm optimization (εICPSO) [Ihara *et al.*, 2019] and ε constrained genetic algorithm (εDGA) [Ihara *et al.*, 2018]. Candidate solutions (assigned segments) are evaluated by the two-dimensional simulator we developed. Solutions are encoded to particles and integer chromosome and they are evolved by the εCCICPSO, εICPSO, and εDGA.

4.1 Fitness and Constraint Evaluation

The simulator evaluates segments by the area of the region a shield machine passed along to the segments assuming that the amount of excavated soils is in proportion to the area. The area of the region through which the front of the shield machine passes is determined by the product of the width of the shield machine and the total length of the planning line. Thus, We define the area of the excavated field excluding this field as the fitness because this field does not depend on segment assignment. This fitness is equivalent to the amount of the soil excavated by the overcut.

4.2 Comparative Methods

Conventional Method

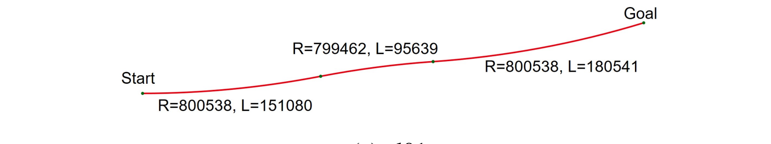
In the construction site, segments are manually assigned by skilled engineers. However it is difficult to compare with real skilled engineers' assignment, because engineers take a lot of time to assign segments in each problem. Since skilled engineers assign in order to minimize the gaps without considering the amount of excavated soil, their methods are approximately equivalent to the greedy method where segments are assigned to minimize gaps, as shown in Algorithm 1. Thus we compare the proposed methods to the greedy method instead of skilled engineers.

εICPSO and εDGA

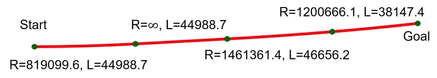
For the segment assignment problem, Ihara *et al.* proposed the εICPSO [Ihara *et al.*, 2019] and εDGA [Ihara *et al.*, 2018], where the ε constraint method is adapted to the ICPSO and discrete genetic algorithm. The εDGA is basically based on standard genetic algorithms, but in the algorithm, individuals are ranked by the ε level comparison with the ε level controlled in each generation. In particular, the parents are selected by selection methods based on comparison of individuals such as tournament selection [Miller *et al.*, 1995] and ranking selection [Goldberg and Deb, 1991] using the ε level comparison instead of general comparison. Elite individuals are also selected by ε level comparison to carry over to the next generation according to elitism. Algorithm 2 summarizes the εDGA.

4.3 Experimental Setup

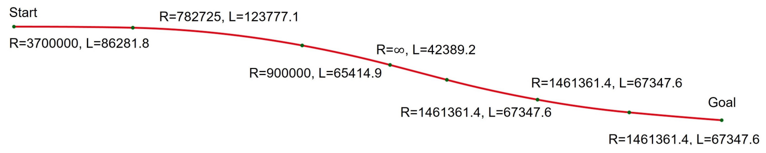
The algorithms tackle the constrained combinatorial optimization problem defined in (1). Planning lines are defined by the given series of curvature radius R and length L. Fig. 3a, 3b, and 3c show the planning lines used in the experiment `pl01`, `pl02`, and `pl03`. The segments are defined as shown in Fig. 3d, and we use the segment sets `sg01` and `sg02`, which include segments whose type number is from one to three,



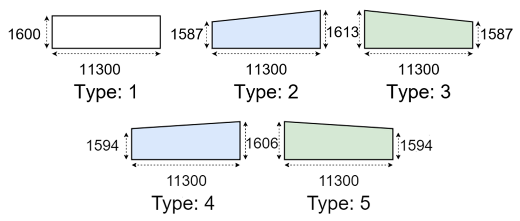
(a) pl01



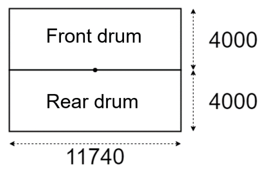
(b) pl02



(c) pl03



(d) segments



(e) shield machine

Figure 3: Dimensions [mm] of the planning lines, the segments, and the shield machine used in the experiments.

Algorithm 1 Greedy Method for Segment Assignment

```
1: Decision variable $\boldsymbol{x} = (x_1, \ldots, x_n)^T$, $x_i \in \{1, \ldots, k\}$
2: $initialize\_position(P_0)$
3: for $i = 1$ to $n$
4:     for $j = 1$ to $k$
5:         $P_i^j \leftarrow next\_position(P_{i-1}, Segment_j)$
6:         $g_j \leftarrow gap(P_i^j)$
7:     end for
8:     $x_i \leftarrow \arg\min_{j\in\{1,\ldots,k\}} g_j$
9:     $P_i \leftarrow P_i^{x_i}$
10: end for
11: return $\boldsymbol{x}$
```

Algorithm 2 εDGA

```
1: $initialize\_population(P)$
2: $evaluate\_population(P)$
3: while (termination condition not met)
4:     $\varepsilon \leftarrow control\_epsilon\_level()$
5:     $P_{mating} \leftarrow selection(P_t, \varepsilon)$
6:     while ($P_{mating}$ is not empty)
7:         $parents \leftarrow pop(P_{mating})$
8:         $offspring \leftarrow crossover(parents)$
9:         $offspring \leftarrow mutate(offspring)$
10:        add $offspring$ to $P_{offspring}$
11:    end while
12:    $P_{offspring} \leftarrow evaluate(P_{offspring})$
13:    $P \leftarrow replacement(P, P_{offspring}, \varepsilon)$
14: end while
15: return the best individual ranked by $<_0$
```

and from one to five, respectively. The gap tolerance g_t is set to $50\,\mathrm{mm}$ in each problem.

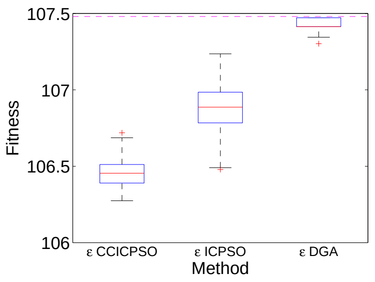
We conduct 50 trials of evolutions where fitness evaluations are limited up to 500,000 times with all of the algorithms. The εCCICPSO decomposes the original problem to 5 sub-components, and uses 5 swarms of size 50. The swarms are evolved for $2,000$ iterations. The εICPSO uses a swarm of size 100, and the swarm is evolved for $5,000$ iterations. They are owing to the recommendation of [Engelbrecht, 2014], which demonstrated that a large swarm may, counterintuitively, have difficulty exploring the search space. In the both of εCCICPSO and εICPSO, the cognitive component ϕ_1 and social component ϕ_2 are set to 1.49618, and the inertia ω is 0.729, which has been found to encourage convergent trajectories [Eberhart and Shi, 2000]. εCCICPSO uses the scaling factor $\epsilon = 0.1$. In the `pl01` and `pl02` problems, εICPSO uses the scaling factor $\epsilon = 5.0 \times 10^{-4}$, and in the `pl03` problems $\epsilon = 1.0 \times 10^{-4}$, due to large dimensions of the problems. In the εDGA, populations are evolved for 500 generations, with a population of size $1,000$. Uniform crossover [Syswerda, 1989] is applied 95% of the time offspring are produced, and each offspring does uniform mutation [Goldberg, 1989] where each gene has a 5% chance of change to random value. Through the evolutions, ε level is set to 0.

4.4 Experimental Results

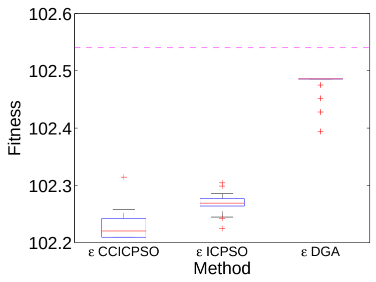
The experimental results show that the proposed method have a potential to find the segment assignment reducing the amount of excavated soil as compared to the conventional method (skilled engineer) while keeping the all gaps between segments and the planning line falling within the tolerance. Because all the algorithms obtained feasible solutions in each trials, only fitness scores are shown. Table 1 shows the experimental results on the problems (Fig. 3). In the table, ”Mean” and “Std dev” are the mean value and the standard deviation value of 50 trials on the each problem respectively. Bold values indicate algorithms that statistically significantly outperformed all other methods (one-tailed Welch’s t-test, $\alpha = 0.01$). Fig. 4 illustrates the performance of the εCCICPSO, the εICPSO and εDGA. Their fitness scores are shown as box plots, where the boxes represent the 25th to 75th percentiles, the lines within the boxes represent the median, and the lines outside the boxes represent the minimum

Table 1: Fitness Scores of Feasible Solutions Obtained by Each Algorithm [m^2] (50trials)

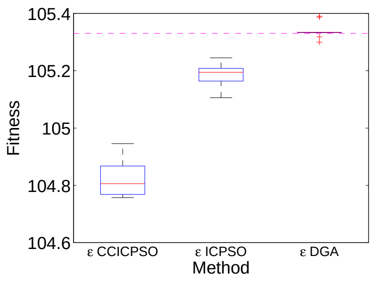
Problem		Skilled engineer	εCCICPSO	εICPSO	εDGA
Segment set	Planning line	(Greedy method)	Mean (Std dev)	Mean (Std dev)	Mean (Std dev)
sg01 ($k = 3$)	pl01 (n=267)	107.48	**106.46 (9.37e-2)**	106.87 (1.54e-1)	107.42 (4.33e-2)
	pl02 (n=210)	102.54	**102.23 (2.05e-2)**	102.26 (1.41e-2)	102.47 (1.82e-2)
	pl03 (n=325)	105.33	**104.81 (5.38e-2)**	105.18 (3.77e-2)	105.33 (1.64e-2)
sg02 ($k = 5$)	pl01 (n=267)	107.35	**106.48 (1.01e-1)**	106.84 (7.30e-2)	107.32 (3.05e-2)
	pl02 (n=210)	102.52	**102.22 (5.95e-3)**	102.25 (1.72e-2)	102.46 (2.55e-2)
	pl03 (n=325)	105.08	**104.85 (2.46e-2)**	104.93 (2.46e-2)	105.08 (1.60e-2)



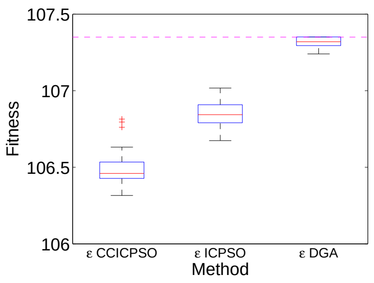
(a) sg01$\times$ pl01



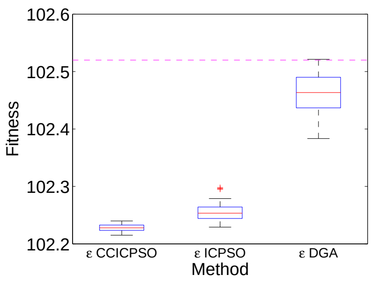
(b) sg01$\times$ pl02



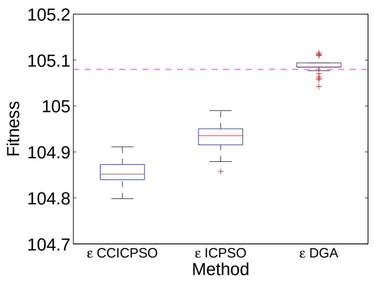
(c) sg01$\times$ pl03



(d) sg02$\times$ pl01



(e) sg02$\times$ pl02



(f) sg02$\times$ pl03

Figure 4: Box plots of fitness scores of all the algorithms for each problem with 50 trials, with horizontal dashed lines representing the conventional method's evaluations.

and maximum values. The conventional method's scores are represented by the horizontal dashed lines.

It is clear that the εCCICPSO has clear advantage over the other algorithms. In all the problems, the εCCICPSO statistically performs the best. In particular, the worst scores of εCCICPSO exceed the εICPSO's average, εDGA's best and the skilled engineer's score. In complex problems, with large n or k, the difference in the performance is especially remarkable. Although the εDGA has potential to find the solution superior to the skilled engineer in terms of the best score, its score averagely almost equivalent and at worst inferior in the pl03 problems.

5 Conclusion

We addressed the segment assignment in shield tunneling as a constrained combinatorial optimization problem. This paper proposed the εCCICPSO and demonstrated its effectiveness to segment assignment problems. The experimental results showed its potential to reduce construction costs as compared with the conventional method. In all the test problems, the proposed method outperformed all the comparative methods. In the future, we will make more experiments using three-dimensional simulator for more accurate evaluation of the proposed method.

Acknowledgements

This work was supported in part by the Ministry of Education, Science, Sports and Culture, Grant–in–Aid for Scientific Research under grant #JP19H01137, #JP19H04025, and #JP20H04018.

References

[Chen *et al.*, 2019] Ren-Peng Chen, Pin Zhang, Xin Kang, Zhi-Quan Zhong, Yuan Liu, and Huai-Na Wu. Prediction of maximum surface settlement caused by earth pressure balance (EPB) shield tunneling with ANN methods. *Soils and Foundations*, 59(2):284–295, 2019.

[Eberhart and Shi, 2000] Russell C. Eberhart and Yuhui Shi. Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, volume 1, pages 84–88. IEEE, 2000.

[Engelbrecht, 2014] Andries Petrus Engelbrecht. Fitness function evaluations: A fair stopping condition? In *Proceedings of 2014 IEEE Symposium on Swarm Intelligence (SIS)*, pages 1–8. IEEE, 2014.

[Goldberg and Deb, 1991] David E. Goldberg and Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms (FOGA)*, volume 1, pages 69–93. Elsevier, 1991.

[Goldberg, 1989] David E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, Reading, MA, 1989.

[Hasanipanah *et al.*, 2016] Mahdi Hasanipanah, Majid Noorian-Bidgoli, Danial Jahed Armaghani, and Hossein Khamesi. Feasibility of PSO-ANN model for predicting surface settlement caused by tunneling. *Engineering with Computers*, 32(4):705–715, 2016.

[Hassanien and Emary, 2018] Aboul Ella Hassanien and Eid Emary. *Swarm intelligence: principles, advances, and applications*. CRC Press, 2018.

[Ihara *et al.*, 2018] Koya Ihara, Shohei Kato, Takehiko Nakaya, and Tomoaki Ogi. Constrained GA based segment assignment in shield tunneling to minimize the amount of excavated soil. In *Proceedings of 2018 IEEE 7th Global Conference on Consumer Electronics*, pages 229–230. IEEE, 2018.

[Ihara *et al.*, 2019] Koya Ihara, Shohei Kato, Takehiko Nakaya, Tomoaki Ogi, and Hiroichi Masuda. Application of PSO-based constrained combinatorial optimization to segment assignment in shield tunneling. In *Agents and Artificial Intelligence*, pages 166–182. Springer International Publishing, 2019.

[Japan Society of Civil Engineers, 2007] Japan Society of Civil Engineers. *Standard specifications for tunneling-2006: shield tunnels*. Tunnel Engineering Committee, 8 2007.

[Kennedy and Eberhart, 1995] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942–1948. IEEE, 1995.

[Koyama, 2003] Yukinori Koyama. Present status and technology of shield tunneling method in japan. *Tunnelling and Underground Space Technology*, 18(2-3):145–159, 2003.

[Larrañaga and Lozano, 2002] Pedro Larrañaga and Jose A Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer Science & Business Media, 2002.

[Lin *et al.*, 2019] Xing-Tao Lin, Ren-Peng Chen, Huai-Na Wu, and Hong-Zhan Cheng. Deformation behaviors of existing tunnels caused by shield tunneling undercrossing with oblique angle. *Tunnelling and Underground Space Technology*, 89:78–90, 2019.

[Maidl *et al.*, 2013] Bernhard Maidl, Martin Herrenknecht, Ulrich Maidl, and Gerhard Wehrmeyer. *Mechanised shield tunnelling*. John Wiley & Sons, 2013.

[Miller *et al.*, 1995] Brad L Miller, David E. Goldberg, et al. Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3):193–212, 1995.

[Ng and Chan, 2015] Jacky YK Ng and Alan HS Chan. The ageing construction workforce in Hong Kong: A review. In *Proceedings of International MultiConference of Engineers and Computer Scientists 2015 (IMECS)*. Newswood Limited, 2015.

[Olsen *et al.*, 2012] Darren Olsen, Mark Tatum, and Christopher Defnall. How industrial contractors are handling skilled labor shortages in the United States. In *48th ASC Annual International Conference Proceedings*, 2012.

[Pampara *et al.*, 2005] Gary Pampara, Nelis Franken, and Andries Petrus Engelbrecht. Combining particle swarm optimisation with angle modulation to solve binary problems. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, volume 1, pages 89–96. IEEE, 2005.

[Pitakaso *et al.*, 2020] Rapeepan Pitakaso, Kanchana Sethanan, and Thitipong Jamrus. Hybrid pso and alns algorithm for a software and mobile application for transportation in the ice manufacturing industry. *Computers & Industrial Engineering*, page 106461, 2020.

[Potter and Jong, 2000] Mitchell A Potter and Kenneth A De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary computation*, 8(1):1–29, 2000.

[Singh and Salgotra, 2019] Urvinder Singh and Rohit Salgotra. Synthesis of linear antenna arrays using enhanced firefly algorithm. *Arabian Journal for Science and Engineering*, 44(3):1961–1976, 2019.

[Strasser *et al.*, 2016] Shane Strasser, Rollie Goodman, John Sheppard, and Stephyn Butcher. A new discrete particle swarm optimization algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 53–60. ACM, 2016.

[Suzuki, 2016] Atsushi Suzuki. 2016 annual report of NILIM: Productivity improvement in infrastrucure development process using i-construction. http://www.nilim.go.jp/english/annual/annual2016/ar2016e.html, 2016. Accessed: 2018-10-23.

[Syswerda, 1989] Gilbert Syswerda. Uniform crossover in genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 2–9. Morgan Kaufmann Publishers, 1989.

[Takahama and Sakai, 2005] Tetsuyuki Takahama and Setsuko Sakai. Constrained optimization by ε constrained particle swarm optimizer with ε-level control. In *Soft Computing as Transdisciplinary science and technology*, pages 1019–1029. Springer, 2005.

[Wiegand *et al.*, 2001] R Paul Wiegand, William C Liles, and Kenneth A De Jong. An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pages 1235–1242. Morgan Kaufmann Publishers Inc., 2001.

[Zhang *et al.*, 2014] Zhongshan Zhang, Keping Long, Jianping Wang, and Falko Dressler. On swarm intelligence inspired self-organized networking: its bionic mechanisms, designing principles and optimization approaches. *IEEE Communications Surveys & Tutorials*, 16(1):513–537, 2014.

[Zhang *et al.*, 2020] WG Zhang, HR Li, CZ Wu, YQ Li, ZQ Liu, and HL Liu. Soft computing approach for prediction of surface settlement induced by earth pressure balance shield tunneling. *Underground Space*, 2020.

[Zhou *et al.*, 2019] Cheng Zhou, Hengcheng Xu, Lieyun Ding, Linchun Wei, and Ying Zhou. Dynamic prediction for attitude and position in shield tunneling: A deep learning method. *Automation in Construction*, 105:102840, 2019.