The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

εCCICPSO を用いたシールド工法におけるセグメント割当

Segment Assignment in Shield Tunneling Using εCCICPSO

伊原 滉也 *1 *2　加藤 昇平 *1 *2
Koya Ihara　Shohei Kato

*1 名古屋工業大学 大学院工学研究科情報工学専攻
Dept. of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

*2 名古屋工業大学 情報科学フロンティア研究院
Frontier Research Institute for Information Science, Nagoya Institute of Technology

Recent years, segment assignment problems (SAPs) are under intense investigation in order to improve productivity in the shield tunneling construction. In previous study, the authors proposed the ε constrained integer categorical particle swarm optimization (εICPSO) for SAPs, and demonstrated its effectiveness to SAPs. This paper introduced the cooperative coevolutionary framework to the εICPSO for large-scale SAPs. The proposed method statistically outperformed all the comparative methods in the simulation experiments using real construction data.

1. はじめに

シールド工法 [土木 16] は世界中で広く用いられているトンネル工法の 1 つで，都市部等の厳しい条件下でのトンネル建設に必要不可欠な工法である．現在シールド工事の施工計画時に技術者によって行われているセグメント割付は，自動化と最適化による省人化と施工コストの大幅な削減が見込まれ，制約付組合せ最適化問題として近年盛んに研究されている [Ihara 19a, Ihara 19b].

Ihara・Kato は，ε 制約法 [Takahama 05] を離散 GA に導入した ε constrained discrete genetic algorithm (εDGA) を提案し，シミュレーション実験により従来の技術者の割付と比較して掘削コストを削減できることを実証した [Ihara 19a]．また Ihara・Kato・Nakaya・Ogi・Masuda は，組合せ最適化のために提案された PSO の一種である Integer Categorical Particle Swarm Optimization (ICPSO) [Strasser 16] と ε 制約法を組合せた εICPSO を開発し，実験における全問題で εDGA のスコアを統計的に有意に上回った [Ihara 19b]．しかし，次元数の大きい問題設定では掘削コストを大きく削減するには至らなかった．

次元数の大きい問題に取組む上では分割統治法がよく用いられている．分割統治の考え方を進化計算に導入したフレームワークとして Cooperative Coevolution (CC) [Potter 00] が提案されている．本稿では εICPSO の更なる改良のため CC を導入し，そのセグメント割付に対する有効性を実データを用いたシミュレーション実験によって検証する．

2. セグメント割付計画問題

セグメント割付計画では，図 1 のように直線と曲線からなるトンネルの計画線（planning line）に沿って，工事ごとに定められた複数タイプのセグメント（segment）を割付ける．セグメント割付には次の 2 つの要求がある．

- 割付た全てのセグメントと計画線とのズレを許容値 g_t 以内に収めること．

連絡先: 加藤昇平，名古屋工業大学，名古屋市昭和区御器所町，052-735-5625，shohey@katolab.nitech.ac.jp

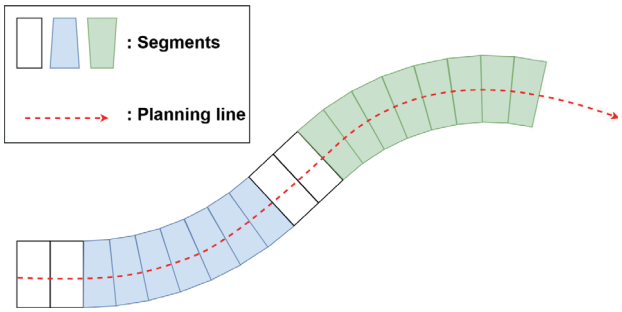


図 1: An Example of Segment Assignment

- 割付けたセグメントを通って掘削するシールドマシンの掘削土量を少なくすること．

前者を不等式制約，後者の掘削土量を目的関数として，次のような最適化問題として定義される．

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

ここで，$x_i \in \{1, \cdots, k\}$ は i 番目に割付られたセグメントのタイプを表し，決定変数ベクトル $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ は割付けたセグメントの系列を表す．使用するセグメントは k タイプあるものする．また，$f(\boldsymbol{x})$ はセグメント列 $\boldsymbol{x}$ を配置する際にシールドマシンが掘削する土量，g_t はズレの許容値，$g_i(\boldsymbol{x})$ はセグメント列 $\boldsymbol{x}$ において i 番目のセグメントの計画線とのズレを表す．ここでいうズレとは，セグメントの中心点と計画線のユークリッド距離である．

3. 提案手法

本稿では CCICPSO と ε 制約法によるセグメント割付（εCCICPSO）を提案する．この章では CCICPSO と ε 制約法について詳細を述べる．

Algorithm 1 CCICPSO

1: **for** $k = 1$ **to** the number of subcomponents d
2: **for each** particle p_i of the sub-swarm $swarm_k$
3: $initialize_particle(p_i)$
4: $initialize_local_best(p_i)$
5: **end for**
6: $initialize_global_best(swarm_k)$
7: **end for**
8: $initialize_evalutions(swarms)$
9: **while** (termination condition not met)
10: **for** $k = 1$ **to** d
11: **for each** particle p of the sub-swarm $swarm_k$
12: $S_p \leftarrow sample(X_p)$
13: $\boldsymbol{S} \leftarrow$ collaborate S_p with the global best samples of the other swarms
14: $\boldsymbol{S}.evaluation \leftarrow evaluate(\boldsymbol{S})$
15: **if** $\boldsymbol{S}$ is superior to p's local best solution $\boldsymbol{S}_{\mathrm{pBest}_p}$
16: $X_{\mathrm{pBest}_p} \leftarrow update_best_position(X_p, S_p, X_{\mathrm{pBest}_p})$
17: $\boldsymbol{S}_{\mathrm{pBest}_p} \leftarrow \boldsymbol{S}$
18: **end if**
19: **if** $\boldsymbol{S}$ is superior to p's global best solution $\boldsymbol{S}_{\mathrm{gBest}}$
20: $X_{\mathrm{gBest}} \leftarrow update_best_position(X_{p_i}, S_{p_i}, X_{\mathrm{gBest}})$
21: $\boldsymbol{S}_{\mathrm{gBest}} \leftarrow \boldsymbol{S}$
22: **end if**
23: **end for**
24: **for each** particle p_i of the sub-swarm $swarm_k$
25: $V_p \leftarrow update_velocity(X_p, V_p, X_{\mathrm{pBest}_i}, X_{\mathrm{gBest}})$
26: $X_p \leftarrow update_position(X_p, V_p)$
27: **end for**
28: **end for**
29: **end while**
30: **return** combination of the global best samples of all the swarms

3.1 CCICPSO

CCICPSO は Potter・De Jong のモデル [Potter 00] に基づいて，対象の変数ベクトルを分割し部分問題ごとに対応するサブ粒子群を用いて最適化を実行する．それぞれのサブ粒子群は同時にかつ独立に ICPSO によって進化する．

ICPSO は Strasser・Goodman・Sheppard・Butcher によって提案された順序関係のない離散変数の最適化を扱う新しい PSO アルゴリズムであり，名義尺度上の離散最適化問題において他の離散 PSO アルゴリズムよりも性能が上回ることが実証されている [Strasser 16]．PSO アルゴリズムでは，探索空間内の最良の位置を粒子群が探索する．それぞれの粒子は位置と速度を持ち，粒子群全個体の情報を参照して速度と位置の更新を行う．通常，位置と速度は連続値を前提としており，位置そのものが連続値を持つ解候補に相当する．ICPSO では，粒子 p の位置 X_p は決定変数ベクトルを直接表すのではなく，Estimation of Distribution Algorithm (EDA) [Larrañaga 02] のように，要素ごとの取りうる値の確率分布を表す．その確率分布からサンプリングされた解 S_p の適応度を評価し，粒子 p の適応度とする．これによって決定変数が離散値でも，位置と速度は連続値であるため，通常の PSO 同様の位置と速度の更新式を用いて探索を行うことができる．位置と速度は，それまでに発見した最良位置を利用して更新される．ICPSO では粒子 p の位置 X_p からのサンプル $S_p = [s_{p,1}, s_{p,2} \cdots, s_{p,n}]$ がベストサンプルを超えた場合，ベスト位置の確率分布は各変数 X_i のすべての状態 $j \in Vals(X_i)$ に対して次のように更新される．

$$d^j_{\mathrm{Best},i} = \begin{cases} \epsilon d^j_{p,i} & (j \neq s_{p,i}) \\ d^j_{p,i} + \sum\limits_{\substack{k \in Vals(X_i) \\ \wedge k \neq j}} (1-\epsilon) d^k_{p,i} & (j = s_{p,i}) \end{cases} \quad (1)$$

ここでスケーリングファクタ ϵ はユーザが $[0, 1)$ の範囲内で設定するパラメータで，確率分布の変化量を決定する．$d^j_{p,i}$，$d^j_{\mathrm{Best},i}$ はそれぞれ粒子 p，ベストの確率分布において，変数 X_i が値 j をとる確率を表す．

部分解の適応度は他の各粒子群から選択された部分解（コラボレータ）との組み合わせから決定される．コラボレータの選択方法は多数存在する [Wiegand 01] が，本稿では最も広く用いられている各集団から最良個体を選択する手法を用いる．すなわち，CCICPSO ではコラボレータとして各粒子群のグローバルベストサンプルが用いられる．

Algorithm 1 に CCICPSO の全体の流れを示す．分割数を d とし，Step 1-7 で d 個の群れを初期化する．Step 6 ではグローバルベストを初期化する．コラボレータ（グローバルベストサンプル）が定義されていない状態では適応度評価ができないため，ここではランダムに選択された粒子とサンプルをグローバルベストとする．その後 Step 8 で全粒子の適応度評価を行う．Step 9 から 29 では d 個の粒子群をそれぞれ進化させる．Step 16 と 20 は (1) に基づいて粒子 p のローカルベスト位置 X_{pBest} と全群のグローバルベスト位置 X_{gBest} の位置の更新を行う．これらのベスト位置を用いて，Step 25 と 26 で位置と速度を更新する．

3.2 ε 制約法

ε 制約法 [Takahama 05] では，制約付き最適化問題を直接探索法で解く際に，通常の比較の代わりに ε レベル比較を用いる．ε レベル比較は，制約違反の尺度である制約逸脱度と目的関数の評価値の両方を考慮した比較であり，これを用いて解候補を比較することで，制約無し最適化用に設計された既存手法が制約付き最適化問題に対応できるようになる．

制約をどの程度違反しているか量る尺度として制約逸脱度 $\phi(\boldsymbol{x})$ を導入する．本稿では，次の定義を採用した．

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

ε レベル比較は目的関数値と制約逸脱度の組 (f, ϕ) の集合上の比較である．制約逸脱度が ε 以下の場合と制約逸脱度が同値の場合は目的関数値の大小関係，それ以外の場合は制約逸脱度の大小関係を用いて，(f, ϕ) の大小を決定する．具体的には ε レベル比較 $<_\varepsilon$ $(\varepsilon \in (0, \infty))$ は解 $\boldsymbol{x}_1, \boldsymbol{x}_2$ の関数値を f_1, f_2，制約逸脱度を ϕ_1, ϕ_2 として次のように定義される．

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

4. εCCICPSO によるセグメント割付実験

本章では，シールド工事の実データを用いた二次元シミュレーション実験によって，提案手法のセグメント割付に対する有効性を検証する．

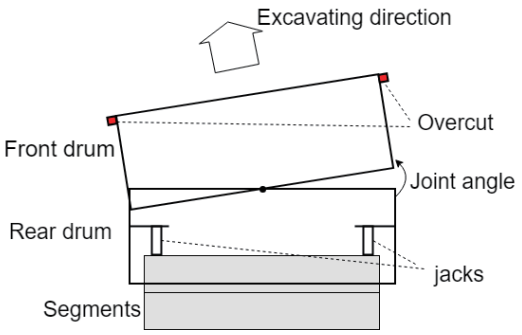


図 2: Model of Shield Machines in a Two-Dimensional Simulator

4.1 シミュレータによる適応度と制約逸脱度評価

解の適応度と制約逸脱度の評価を二次元シミュレータを用いて行う．シミュレータ上では図 2 のようなシールドマシンを考える．掘削土量がシールドマシンの掘削した領域の面積に比例すると仮定すると，掘削土量を最小化するにはマシンの通過領域の面積を最小化すればよい．通過領域は，あるセグメント列が与えられたとき，セグメント 1 つ 1 つに対してそれを配置するシールドマシンの位置を計算し，そのそれぞれの前胴（front drum），後胴（rear drum）の各頂点の軌跡から求めることができる．このとき，前胴の前面が通過する領域の面積は，前胴の幅と計画線の全長の積で求まり，どのようなセグメント配置をしても変化しないため，この領域を除いた領域の面積を適応度とする．これは，図 2 の余堀り（overcut）で掘削した領域の面積に相当する．

制約逸脱度は (2) のように許容値を超えたズレの総和として定義されている．二次元シミュレータ上においてセグメントと計画線のズレは，セグメントの前方の辺の中点から計画線におろした垂線の長さとして定義される．

4.2 比較手法

シールド施工の現場では熟練技術者の手によってセグメントの割付が行われている．二次元に単純化したシミュレータ上において，三次元的に割付ける現実の技術者と直接比較することはできないため，先行研究 [Ihara 19b] で技術者による割付の近似として用いられたズレのみを評価とした貪欲法による手法 (Engineer*) を，技術者の代わりとして比較する．加えて εDGA [Ihara 19a]，εICPSO [Ihara 19b] とも比較する．

4.3 初期個体

制約付最適化において，特に実行可能領域が狭い場合，実行可能領域もしくはその近傍から探索を開始することが望ましい．本実験では実行可能解はシミュレータと前述の Engineer* を用いて発見できるため，獲得した実行可能解 S^* を用いてアルゴリズムを初期化する．εICPSO では，粒子の速度と位置ベクトルは全てランダムに初期化される．そのうち 1 つの粒子のサンプリングから S^* が得られたと仮定して，ローカルベストとグローバルベストの更新を行う．εCCICPSO では，各群について S^* を分割した部分解が得られたと仮定して同様の更新を行う．εDGA では得られた S^* に一様突然変異を適用した解を初期個体として用いる．

4.4 問題設定

文献 [Ihara 19b] と同様に，それぞれ長さや形の違う計画線 `pl01`，`pl02`，`pl03` とセグメントセット `sg01`，`sg02` を用いて計 6 種のセグメント割付実験を行った．`sg01` には 3 タイプ，`sg02` は `sg01` に 2 タイプ加えた計 5 タイプのセグメントが含まれる．それぞれのセグメントで形は違うが長さは一律であるため，計画線の長さによって問題の次元数（割付るセグメント数）が決まり，`pl01`，`pl02`，`pl03` はそれぞれ 267，210，325 次元となる．各問題でセグメントと計画線のズレの許容値は $g_t = 50$ [mm] とした．これらの問題設定は実際のシールド工事で用いられた値から作成した．

4.5 アルゴリズム設定

各アルゴリズムについて 50 試行ずつ実施した．比較のため，それぞれの試行における解候補の評価回数を最大 500,000 回とした．提案手法の分割数 5，群サイズ 50，反復数を 2,000 とした．スケーリングファクタは $\epsilon = 0.01$ とした．これら以外のパラメータは，文献 [Ihara 19b] と同様に設定した．

4.6 実験結果

提案手法はズレを許容値以下に収めたうえで，掘削土量をどの比較手法よりも削減できることが分かった．表 1 に各問題における実験結果を示す．全てのアルゴリズムが各問題の各試行において実行可能解を獲得できたため，最終的な解の適応度のみ抜粋した．適応度は前述の掘削土量をシミュレートした評価値になっている．表内の average，best，worst は各アルゴリズムの 50 試行における適応度の平均値，最小値，最大値を表す．平均値のうち太字で示したものは他の全てのアルゴリズムに対して統計的に有意な差があったことを示す (paired Student t-Test, $\alpha = 0.01$)．図 3 は全アルゴリズムの性能差を図示している．箱ひげ図は εCCICPSO，εICPSO，εDGA の 50 試行における解の評価値の分布を示す．ここで，箱は第 1 四分位点と第 3 四分位点を表し，箱内の線は中央値，箱外の線は最小値と最大値を表している．破線は技術者の割付の近似である Engineer*の評価値を示している．

εCCICPSO は他手法と比較して明らかに性能が上回った．全ての問題設定において，他の全アルゴリズムに対して有意水準 1%で統計的に有意な差が見られた．ほとんどの問題で εCCICPSO の worst が，εICPSO の average を上回っている．特に次元数 n の大きい問題ほど，εCCICPSO と εICPSO の性能差が顕著にみられる．

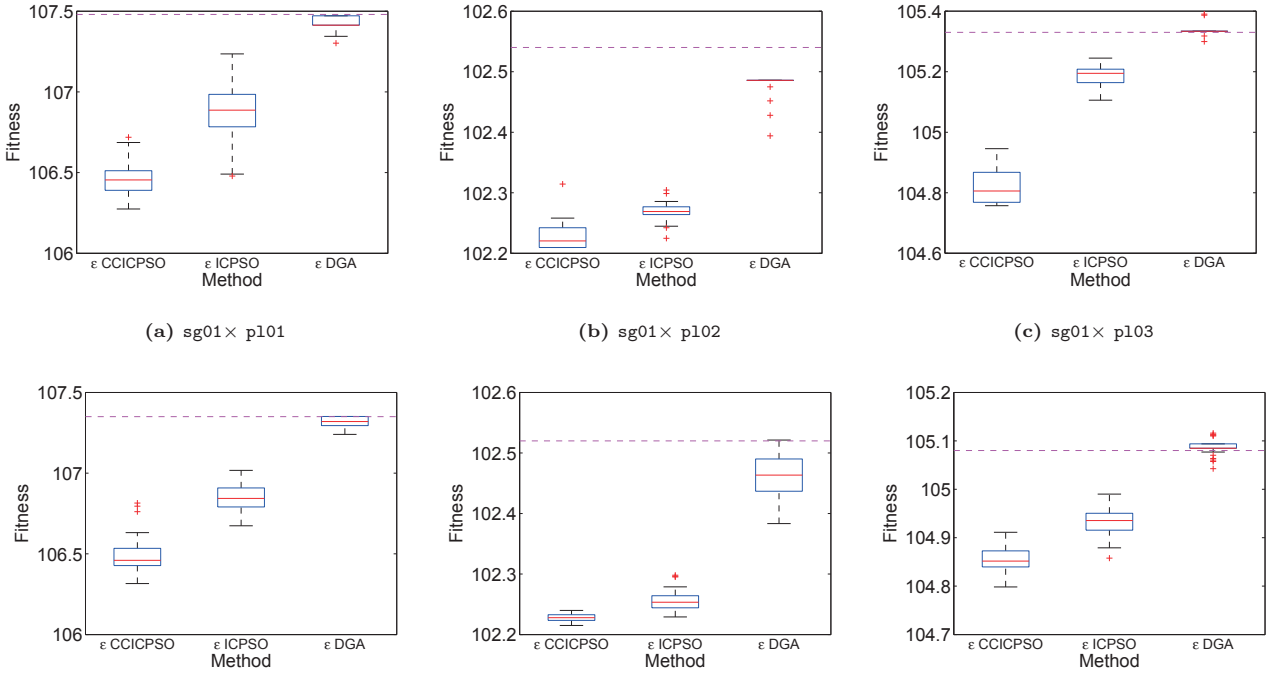
一方で，各計画線ごとに結果を比較すると，εCCICPSO は `sg02` を用いた問題設定では `sg01` を用いた場合に比べて適応度が僅かに悪化していることがわかる．本来 `sg02` は `sg01` のセグメントをすべて含むため，`sg02` を用いた問題には `sg01` の場合と同等かそれ以上に良い解が存在するはずである．実際，εCCICPSO 以外の手法は同手法で `sg01` を用いた場合よりも，同等以上の解を獲得している．これは εCCICPSO において，セグメントタイプ数が大きいほど問題分割による局所収束のデメリットが大きくなっていることが理由として考えられる．分割によるデメリットは問題の変数依存関係を考慮した分割や動的な問題分割によって解消することができるため，今後はセグメント割付問題に有効な問題分割方法を調査していく．

5. おわりに

本稿では，シールド工法におけるセグメント割付計画問題を制約付き組合せ最適化問題として解決した．大規模最適化問題に対する探索能力向上のために εICPSO に共進化を導入し，実データを用いた二次元シミュレーション実験によりそのセグメント割付に対する有効性を確認した．全問題設定において，提案手法は比較手法に対して統計的に有意に上回った．特に，次元数の大きい問題に対する大幅な性能向上が確認できた．一方で，セグメントタイプ数の多い複雑な問題では問題分割によ

表 1: Fitness scores of feasible solutions obtained by each algorithm [m^2] (50trials)

Problem		Engineer*	εCCICPSO			εICPSO			εDGA		
Segment set	Planning line		average	best	worst	average	best	worst	average	best	worst
	pl01 (n=267)	107.48	**106.46**	106.27	106.71	106.87	106.47	107.23	107.42	107.30	107.47
sg01 ($k=3$)	pl02 (n=210)	102.54	**102.23**	102.21	102.31	102.26	102.22	102.30	102.47	102.39	102.48
	pl03 (n=325)	105.33	**104.81**	104.75	104.94	105.18	105.10	105.24	105.33	105.30	105.39
	pl01 (n=267)	107.35	**106.48**	106.31	107.81	106.84	106.67	107.01	107.32	107.24	107.35
sg02 ($k=5$)	pl02 (n=210)	102.52	**102.22**	102.21	102.24	102.25	102.22	102.29	102.46	102.38	102.52
	pl03 (n=325)	105.08	**104.85**	104.79	104.91	104.93	104.85	104.99	105.08	105.04	105.11



(a) sg01× pl01 (b) sg01× pl02 (c) sg01× pl03

(d) sg02× pl01 (e) sg02× pl02 (f) sg02× pl03

図 3: Box plots of fitness scores of all the algorithms for each problem with 50 trials (horizontal dashed lines represent the conventional method's evaluations)

る局所収束が起きやすい可能性が示唆された．今後は問題に適した分割手法の開発や，実用化に向けて三次元シミュレータを用いた実世界により近い環境での評価実験に着手する．

参考文献

[Ihara 19a] Ihara, K. and Kato, S.: A GA Based Approach for Segment Assignment to Reduce Excavated Soil in Shield Tunneling, in *Proceedings of the 24th International Symposium on Artificial Life and Robotics 2019*, pp. 270–274 (2019)

[Ihara 19b] Ihara, K., Kato, S., Nakaya, T., Ogi, T., and Masuda, H.: A PSO Based Approach to Assign Segments for Reducing Excavated Soil in Shield Tunneling, in *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART)*, pp. RP–136 (8pages) (2019), (to appear)

[Larrañaga 02] Larrañaga, P. and Lozano, J. A.: *Estimation of distribution algorithms: A new tool for evolutionary computation*, Vol. 2, Springer Science & Business Media (2002)

[Potter 00] Potter, M. A. and Jong, K. A. D.: Cooperative coevolution: An architecture for evolving coadapted subcomponents, *Evolutionary computation*, Vol. 8, No. 1, pp. 1–29 (2000)

[Strasser 16] Strasser, S., Goodman, R., Sheppard, J., and Butcher, S.: A new discrete particle swarm optimization algorithm, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pp. 53–60ACM (2016)

[Takahama 05] Takahama, T. and Sakai, S.: Constrained optimization by ε constrained particle swarm optimizer with ε-level control, in *Soft computing as transdisciplinary science and technology*, pp. 1019–1029, Springer (2005)

[Wiegand 01] Wiegand, R. P., Liles, W. C., and Jong, K. A. D.: An empirical analysis of collaboration methods in cooperative coevolutionary algorithms, in *Proceedings of the 3rd annual conference on genetic and evolutionary computation*, pp. 1235–1242Morgan Kaufmann Publishers Inc. (2001)

[土木 16] 土木学会：2016 年制定 トンネル標準示方書 [シールド工法編]・同解説, JSCE Publications (2016)