Jurnal Teknik Industri, Vol. 19, No. 2, December 2017, 103-114
ISSN 1411-2485 print / ISSN 2087-7439 online
DOI: 10.9744/jti.19.2.103-114

Implementasi Metode Metaheuristik *Symbiotic Organisms Search* dalam Penentuan Tata Letak Fasilitas Proyek Konstruksi Berdasarkan Jarak Tempuh Pekerja

Doddy Prayogo[1]*, Richard Antoni Gosno[1], Richard Evander[1], Sentosa Limanto[1]

Abstract: This research investigates the performance of a new metaheuristic method called symbiotic organisms search (SOS) for solving the site layout of construction project facilities based on travelling distance. Two practical case studies of facility layout are used to examine the accuracy and consistency of the proposed algorithm. Additionally, three other metaheuristic methods, called particle swarm optimization, artificial bee colony, and teaching–learning-based optimization, are employed as a comparison to the SOS algorithm. The simulation results indicate the superiority of the SOS as well as excellent convergence behavior over the other methods in optimizing the site layout of construction facilities.

Keywords: Facility layout; optimization; metaheuristic; symbiotic organisms search.

Pendahuluan

Penentuan tata letak fasilitas merupakan kunci utama dalam dunia konstruksi dan harus dipertimbangkan sedini mungkin terutama dalam tahap perencanaan. Tujuan utama penentuan tata letak fasilitas adalah untuk mengatur seefisien mungkin letak fasilitas-fasilitas sementara seperti kantor, tempat tinggal pekerja, tempat penyimpanan barang, dan lain-lain, sehingga aktivitas pekerjaan dapat dilakukan lebih leluasa dengan biaya seminimal mungkin. Tata letak fasilitas yang baik sangat penting untuk meningkatkan efektivitas kerja serta meminimalkan jarak tempuh pekerja proyek, terutama pada proyek yang berskala cukup besar (Adrian *et al.* [1]). Umumnya, manajer proyek cenderung menggunakan intuisi dan pengalaman kerja proyek sebelumnya sebagai acuan untuk melakukan merencanakan letak fasilitas proyek. Namun, hal ini tidak menjamin terwujudnya suatu perencanaan tata letak fasilitas proyek yang optimal (Tommelein *et al.* [2]. Hal ini mendorong para peneliti untuk menciptakan sejumlah metode yang berpotensi untuk menghasilkan tata letak fasilitas yang efisien.

Banyak studi yang telah dilakukan untuk mengembangkan metode optimasi untuk menyelesaikan proproblem tata letak fasilitas. Beberapa pendekatan matematis dan heuristic telah dikembangkan di awal untuk menyelesaikan problem tersebut, diantaranya *branch-and-bound* (Simmons [3]), *dynamic programming* (Picard dan Queyranne [4]), and *integer linear programming* (Love dan Wong [5]). Setelah itu, Ravi *et al.* [6] memperkenalkan metode heuristik untuk menyelesaikan salah satu permasalah an tata letak fasilitas. Namun, pendekatan matematis ini seringkali tidak cukup untuk menyelesaikan problem-problem berskala besar sedangkan metode pencarian heuristik seringkali mengalami kendala pada rendahnya kualitas solusi yang dihasilkan akibat terjebak pada *local optima.* Hal ini mendorong meningkatnya upaya-upaya untuk mengembangkan metode-metode optimasi yang lebih akurat dan efisien.

Dalam beberapa tahun terakhir, penggunaan algoritma metaheuristik semakin meningkat bila dibandingkan dengan metode konvensional berbasis matematis karena performanya yang sudah teruji dalam menyelesaikan berbagai macam masalah optimasi yang rumit. Algoritma metaheurisitik memiliki skema pencarian solusi yang terinspirasi dari prinsip-prinsip alamiah yang dikembangkan oleh makhluk hidup, antara lain: evolusi dan seleksi alam yang diadopsi oleh *genetic algorithm* (GA, Holland [7]); interaksi sosial dari sekawanan ikan maupun burung yang ditiru oleh *particle swarm optimization* (PSO, Kennedy dan Eberhart [8]; maupun interaksi koloni semut untuk menemukan sumber makanan yang menginspirasi *ant colony optimization* (ACO, Dorigo *et al.* [9]. Salah satu keuntungan algoritma metaheuristik dibandingkan dengan metode optimasi lain adalah mayoritas skema pencariannya mengijinkan diterimanya solusi yang kurang optimal yang membuat cakupan pencarian solusi menjadi lebih luas yang dapat berujung pada penemuan solusi yang lebih baik. Di sisi lain, algoritma metaheuristik juga dapat menghasilkan kualitas solusi yang cukup baik dengan waktu pencarian yang cenderung relatif cepat.

Berkembangnya penelitian ini memacu munculnya banyak penerapan metode metaheuristik untuk menyelesaikan kasus-kasus optimasi tata letak fasilitas. Li dan Love [10] menggunakan GA untuk mengoptimasi tata letak 11 buah fasilitas. Hasil

[1] Fakultas Teknik Sipil dan Perencanaan, Program Studi Teknik Sipil, Universitas Kristen Petra, Jl. Siwalankerto 121-131, Surabaya 60238, Indonesia.
Email: prayogo@petra.ac.id, m21413042@john.petra.ac.id, m21413048@john.petra.ac.id,leonard@petra.ac.id.

* Penulis korespondensi

eksperimen menunjukkan bahwa GA mampu menghasilkan solusi yang cukup baik dengan jumlah iterasi yang relatif sedikit, yaitu kurang dari 100 iterasi. Zouein *et al.* [11] juga menggunakan GA dalam mengoptimasi tata letak fasilitas proyek dengan tambahan pembatas geometris. Lee *et al.* [12] mengimprovisasi GA untuk menyelesaikan permasalahan tata letak yang fasilitas yang ditempatkan di berbagai macam lantai yang memiliki tambahan batasan berupa dinding dan lorong. Önüt *et al.* [13] menggunakan PSO untuk mencari tata letak fasilitas gudang. Zhang dan Wang [14] mengadopsi PSO untuk menemukan tata letak fasilitas proyek konstruksi yang optimal. Hasil simulasi dari PSO kemudian dibandingkan dengan hasil dari GA, di mana PSO mampu menghasilkan solusi yang lebih optimal dengan jumlah iterasi yang lebih sedikit. Lam *et al.*[15] memperkenalkan ACO untuk menyelesaikan tata letak 13 fasilitas sementara proyek konstruksi. Dari hasil eksperimen, ditunjukkan bahwa ada penghematan biaya sebesar 6,3%~10,8% dari biaya semula. *Tabu search* (TS) digunakan oleh Liang dan Chao [16] untuk problem tata letak fasilitas yang didasarkan terhadap aliran sumber daya dari fasilitas dan hubungan terhadap fasilitas yang berdekatan. Yahya dan Saka [17] mengenalkan *artificial bee colony* (ABC) untuk problem tata letak fasilitas yang harus dioptimasi dengan mempertimbangkan 2 buah fungsi objektif yaitu biaya *handling* serta factor keamanan dan lingkungan. Disamping itu, metode metaheuristik hibrida (Setiawan dan Palit [18]) juga dikembangkan guna mendapatkan solusi tata letak fasilitas yang lebih optimal. Seiring dengan meningkatnya kerumitan dari suatu proyek konstruksi, jumlah fasilitas pendukung sementara yang dibutuhkan cenderung ikut meningkat (Mak *et al.* [19]). Kompleksitas dari masalah tata letak fasilitas akan meningkat secara eksponensial seiring dengan bertambah banyaknya jumlah fasilitas. Sebagai contoh, kemungkinan alternatif solusi yang tersedia untuk 10 fasilitas adalah 10!, setara dengan 3.628.000. Hal ini menyebabkan diperlukannya pengembangan metode optimasi yang lebih maju untuk mengatasi kompleksitas dari masalah tata letak fasilitas.

Baru-baru ini, algoritma metaheuristik baru bernama *symbiotic organisms search* (SOS, Prayogo [20], Cheng *et al.* [21]) menarik perhatian para peneliti karena kemampuannya yang terbukti cukup baik dalam menyelesaikan berbagai permasalahan optimasi yang kompleks dalam berbagai macam bidang [22-26]. SOS mengadopsi pola interaksi simbiosis yang umum ditemukan antar makhluk hidup. Dalam mempertahankan kelangsungan hidupnya, makhluk hidup akan berinteraksi dengan makhluk hidup lainnya dalam bentuk simbiosis. Melalui interaksi itulah makhluk hidup dapat meningkatkan kualitas hidupnya untuk dapat bertahan hidup. Terinspirasi oleh fenomena simbiosis ini, algoritma SOS menggunakan tiga fase utama yaitu *mutualism*, *commensalism*, dan *parasitism*, untuk menemukan solusi terbaik secara efektif dan efisien.

Keberhasilan penerapan SOS sebelumnya pada berbagai masalah optimasi yang rumit melatar belakangi investigasi lebih lanjut terhadap kemampuan algoritma ini optimasi tata letak fasilitas proyek dengan meninjau *traveling distance* minimum dari pekerja. Studi komparatif akan dilakukan dengan membandingkan kinerja SOS dan metode metaheuristik lainnya, yaitu PSO, ABC, dan *teaching–learning-based optimization* (TLBO). Selain menggunakan studi kasus *benchmark*, penelitian ini juga menggunakan studi kasus baru yang dikembangkan dari salah satu proyek konstruksi bangunan bertingkat di Surabaya.

Metode Penelitian

Perumusan Masalah Optimasi Tata Letak Fasilitas Proyek Konstruksi

Perencanaan tata letak fasilitas terdiri dari beberapa proses diantara lain mengindentifikasi fasilitas-fasilitas yang dibutuhkan untuk mendukung pelaksanaan konstruksi, menentukan dimensi serta bentuk fasilitas, dan menempatkan fasilitas-fasilitas tersebut dalam suatu ruang terbatas yang tersedia dalam area proyek [2]. Untuk dapat menghasilkan penempatan fasilitas-fasilitas sementara yang efisien, maka perlu dilakukan optimasi pada perencanaan tata letak fasilitas. Pada umumnya, optimasi tata letak fasilitas proyek konstruksi dilakukan untuk mencari penempatan lokasi fasilitas sementara yang menghasilkan biaya termurah. Terdapat beberapa faktor pembatas pada permasalahan tata letak fasilitas, yaitu sejumlah m fasilitas harus diposisikan pada sejumlah n lokasi pada suatu site proyek, dimana $n \geq m$.

Pada penelitian ini, tujuan dari perencanaan tata letak fasilitas adalah untuk meminimalkan jarak tempuh pekerja proyek konstruksi antar fasilitas. Penelitian ini menggunakan asumsi setiap lokasi harus ditugaskan hanya dengan satu fasilitas, dan setiap fasilitas harus ditugaskan di satu lokasi saja. Problem optimasi tata letak fasilitas dapat diformulasikan sebagai *quadratric assignment problem* [27, 28] sebagai berikut.

$$\text{Minimum } TD = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ik} d_{jl} x_{ij} x_{kl} \quad (1)$$

Kendala

$$\sum_{j=1}^{n} x_{ij} = 1, \ i = 1, \dots, n \quad (2)$$

$$\sum_{i=1}^{n} x_{ij} = 1, \ j = 1, \dots, n \quad (3)$$

$$x \in \{0,1\}, i, j = 1, \dots, n \quad (4)$$

dimana n adalah jumlah fasilitas, x_{ij} adalah variabel

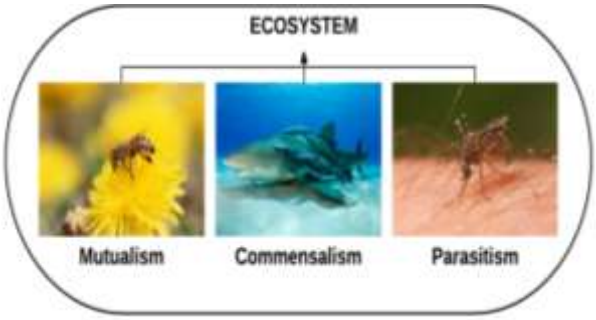
penugasan (x_{ij}= 1 jika fasilitas i ditugaskan ke lokasi j, dan x_{ij}= 0 lainnya). f_{ik} adalah frekuensi perpindahan dalam satu satuan waktu yang dilakukan oleh pekerja proyek konstruksi antara fasilitas i dan k. Dalam penelitian ini satuan waktu yang digunakan adalah satu hari. d_{jl} adalah jarak antara lokasi j dan l. Apabila kedua lokasi berada di samping satu sama lain, jaraknya didefinisikan sebagai jarak antara pusat kedua lokasi; jika tidak, maka jaraknya didefinisikan sebagai sejumlah jarak segmental fasilitas-fasilitas yang ada di antara keduanya (sebagai contoh, d_{13} merupakan total jarak d_{12} dan d_{23}).

Pada penelitian ini, metode metaheuristik yang digunakan dalam proses optimasi akan dijabarkan lebih lanjut. Semua algoritma yang digunakan di sini merupakan algoritma *continuous* berbasis *multi-agent*. Algoritma-algoritma ini memulai proses optimasi dengan satu set alternatif solusi dari problem tersebut dan akan berusaha untuk meningkatkan kualitas dari alternatif solusi tersebut berdasarkan dari fungsi objektif yang diketahui. Berdasarkan serangkaian aturan yang sederhana yang sebagian besar terinspirasi dari alam, algoritma metaheuristik akan memodifikasi solusi yang sudah ada secara iterative dengan tujuan untuk meningkatkan nilai objektif mereka. Bagian selanjutnya akan menjabarkan secara singkat metode SOS bersamaan dengan tiga metode pembanding lainnya yaitu PSO, ABC, dan TLBO.

Symbiotic Organisms Search (SOS)

Algoritma SOS pertama kali diperkenalkan oleh Cheng dan Prayogo [21] untuk menyelesaikan problem berbasis *continuous*. SOS merupakan salah satu algoritma metaheuristik yang mensimulasikan interaksi-interaksi simbiosis yang berbeda-beda yang dilakukan oleh sepasang organisme didalam suatu ekosistem. Interaksi antar organisme dimana masing-masing organisme mendapatkan keuntungan disebut dengan simbiosis mutualisme. Interaksi antar organisme dimana ada satu organisme yang diuntungkan sedangkan yang lainnya netral dikenal sebagai simbiosis komensalisme. Interaksi antar organisme dimana ada satu organisme yang diuntungkan sedangkan yang lainnya dirugikan disebut dengan simbiosis parasitisme. Ketiga macam interaksi simbiosis ini, seperti yang terlihat pada Gambar 1, menginspirasi proses optimasi dari algoritma SOS.

Berbeda dari GA dan metaheuristik berbasis *evolutionary algorithm*, SOS tidak memproduksi atau menciptakan keturunan. Namun, layaknya algoritma yang berbasis populasi, SOS menciptakan sebuah populasi mula-mula (yang disebut dengan ekosistem) dan akan melalui berbagai operator pencarian yang secara iteratif akan mencoba untuk memodifikasi populasi untuk menghasilkan variabel solusi (yang disebut dengan organisme) yang optimal.



Gambar 1. Ilustrasi simbiosis yang terjadi antar makhluk hidup

Operator pencarian dari algoritma SOS terdiri atas tiga jenis, yaitu: *mutualism phase*, *commensalism phase*, dan *parasitism phase*. Uraian mengenai ketiga fase yang disebutkan sebelumnya dijabarkan melalui sub-bab berikut ini.

Mutualism Phase

Pada fase ini, organisme ke-i di dalam ekosistem, O_i, akan berinteraksi secara mutualisme dengan organisme lain yang dipilih secara acak, O_j, dengan tujuan untuk peningkatan kualitas hidup masing-masing di ekosistem. Selanjutnya, dua variabel solusi, new_O_i dan new_O_j akan diciptakan melalui hasil modifikasi setelah simbiosis mutualisme dilakukan yang dimodelkan dengan operasi matematis pada Persamaan (5) dan (6) yang juga melibatkan O_{best}, organisme dengan nilai objektif terbaik dalam ekosistem. Apabila nilai objektif dari kandidat solusi new_O_i dan new_O_j hasil modifikasi O_i dan O_j lebih optimal daripada sebelumnya, organisme O_i dan O_j akan diperbarui.

$$new_{O_i} = O_i + rand(0,1) * (O_{best} - [average(O_i, O_j)] * round[1 + rand(0,1)]) \quad (5)$$

$$new_{O_j} = O_j + rand(0,1) * (O_{best} - [average(O_i, O_j)] * round[1 + rand(0,1)]) \quad (6)$$

Commensalism Phase

Pada fase ini, organisme O_i akan berinteraksi secara komensalisme dengan organisme lain yang dipilih secara acak, O_j. Kali ini, organisme O_i mengambil keuntungan dari interaksi dengan organisme O_j namun organisme O_j sendiri tidak diuntungkan maupun dirugikan. Variabel solusi new_O_i diciptakan melalui hasil modifikasi setelah simbiosis komensalisme dilakukan yang dimodelkan dengan operasi matematis Persamaan (7). Apabila nilai objektif dari kandidat solusi new_O_i hasil modifikasi

O_i lebih optimal daripada sebelumnya, organisme O_i akan diperbarui.

$$new\_O_i = O_i + rand(-1,1) * (O_{best} - O_j) \quad (7)$$

Parasitism Phase

Fase ini digambarkan melalui hubungan antara nyamuk anopheles yang menyebarkan parasit plasmodium ke dalam tubuh inangnya yaitu manusia. Parasit mengambil keuntungan dari hubungannya dengan manusia karena dapat berkembang biak di dalam inangnya. Di sisi lain, manusia dirugikan oleh penyakit malaria yang dibawa melalu parasit tersebut. Fase ini dimulai dengan organisme O_i yang memproduksi parasit buatan bernama "Parasite_Vector". Variabel solusi Parasite_Vector tersusun dari hasil perpaduan antara kloning terhadap organisme O_i dan variabel acak. Organisme O_j, yang berfungsi sebagai inang, dipilih secara acak dari ekosistem. Evaluasi terhadap nilai objektif akan dilakukan terhadap Parasite_Vector dan O_j. Parasite_Vector akan menggantikan posisi organisme O_j di ekosistem hanya jika nilai objektifnya lebih baik. Sebaliknya, O_j akan bertahan dari Parasite_Vector bila nilai objektifnya lebih baik.

$$Parasite\_Vector = \begin{cases} O_{i,k} & r_k < rand(0,1),\ k = 1,2,3,\dots,nvar \\ \text{a random position} & \text{otherwise} \end{cases} \quad (8)$$

di mana *nvar* merupakan jumlah elemen dalam variabel solusi, $O_{i,k}$ menunjukkan nilai variabel dari elemen ke-k dari organisme ke-i dalam ekosistem.

Pada masing-masing operator, para organisme yang ada didalam ekosistem akan berinteraksi secara acak antar satu dengan yang lain. Secara garis besar, algoritma SOS dapat dijabarkan seperti yang terlihat pada Gambar 2.

***Particle Swarm Optimization* (PSO), *Artificial Bee Colony* (ABC), dan *Teaching–Learning-Based Optimization* (TLBO)**

Dikembangkan pertama kali oleh Kennedy dan Eberhart [8], PSO merupakan salah satu metode optimasi metaheuristik tertua yang mengadopsi prinsip *swarm intelligence* atau kecerdasan sosial berkelompok antar makhluk hidup. Mulanya, partikel-partikel akan diciptakan tersebar secara acak dalam lokasi pencarian. Posisi dari partikel-partikel tersebut melambangkan variabel solusi pada problem. Dalam setiap iterasi, partikel-partikel ini akan bergerak sesuai dengan vektor kecepatan dan akan selalu diperbaharui melalui suatu operator matematis yang memodelkan hubungan sosial berkelompok dari grup tersebut seperti yang ditunjukkan dalam Persamaan (9). Bilamana suatu partikel tersebut berhenti pada lokasi baru yang lebih optimal, maka posisi dan nilai objektif dari lokasi suatu partikel tersebut akan disimpan sebagai *pBest* (*personal best*). Selain itu, pada setiap iterasi, akan ada mekanisme untuk menyimpan lokasi global terbaik di antara sekelompok partikel ke dalam *gBest* (*global best*). Secara garis besar, algoritma PSO dapat dijabarkan dalam Gambar 3.

$$v_i = w * v_i + rand(0,1) * c_1 * (x_{pBest} - x_i) + rand(0,1) * c_2 * (x_{gBest} - x_i) \quad (9)$$

di mana v_i merupakan kecepatan dari partikel ke-i, w merupakan parameter *inertia weight*, c_1 merupakan parameter *cognitive factor*, c_1 merupakan parameter *social factor*, x_i adalah koordinat posisi dari partikel ke-i yang melambangkan kandidat solusi dari suatu permasalahan, x_{pBest} adalah koordinat posisi dari *pBest*, x_{gBest} adalah koordinat posisi dari *gBest*.

Sebagai salah satu algoritma berbasis *swarm intelligence* yang ditemukan oleh Karaboga dan Basturk [29], ABC mengadopsi perilaku berkelompok dari lebah dalam mengumpulkan makanan. Mula-mula, algoritma ABC diawali dengan inisialisasi sumber makanan (alternatif solusi) yang berisikan variabel acak. Setelah sumber makanan ditentukan, algoritma akan memasuki tahap pertama, yaitu *employed bees*. Pada tahap ini, *employed bees* akan melakukan modifikasi terhadap alternatif solusi dengan mencari solusi lain di sekitarnya. Solusi tersebut kemudian akan diukur nilai objektifnya sebagai informasi yang akan dibagikan dengan *onlooker bees* melalui *waggle dance* yang dimodelkan melalui persamaan matematis yang dapat dilihat pada Persamaan (10). Pada tahap *onlooker bees*, solusi yang dihasilkan oleh *employed bees* akan dipilih secara acak dengan probabilitas tertentu. *Onlooker bees* kemudian akan memodifikasi kembali solusi tersebut berdasarkan informasi dari *employed bees*. Oleh karena *onlooker bees* memiliki kecenderungan untuk memilih alternatif solusi yang nilai objektifnya lebih baik, maka akan dihasilkan alternatif solusi yang dalam selang waktu tertentu tidak terpilih oleh *onlooker bees*. Pada kondisi ini, *employed bees* akan berubah menjadi *scout bees* untuk mencari alternatif solusi yang baru. Secara garis besar, algoritma ABC dapat dijabarkan dalam Gambar 4.

$$new\_X_i = X_i + rand(-1,1) * (X_i - X_j) \quad (10)$$

di mana X_i merupakan sumber makanan ke-i, dan X_j merupakan sumber makanan ke-j yang dipilih secara acak.

Ditemukan pertama kali oleh Rao, Savsani dan Vakharia [30], TLBO mengambil inspirasi dari

proses transfer ilmu di antara guru dan para muridnya di kelas. Ada jenis dua fase penyusun algoritma TLBO yaitu *teacher phase* dan *learner phase*. Di dalam *teacher phase*, seorang individu akan ditugaskan sebagai *teacher* ($X_{teacher}$) apabila dia memiliki nilai terbaik diantara murid-murid (*learner*) yang lain. TLBO akan berusaha untuk memperbarui individu-individu yang ada dengan mengarahkan nilai rata-rata dari satu kelas (X_{mean}) menuju ke arah $X_{teacher}$. Proses modifikasi solusi yang terjadi di dalam fase ini dimodelkan dalam Persamaan (11). Di dalam *learner phase*, transfer ilmu akan dilanjutkan dengan proses diskusi dan interaksi antara dua orang *learner* (X_i dan X_j) dengan dengan tujuan untuk meningkat nilai dari masing-masing individu. Proses modifikasi solusi yang terjadi di dalam fase ini dimodelkan dalam Persamaan (12). Dalam hal memperbaharui kualitas solusi dari individu, TLBO juga menerapkan konsep *greedy selection*. Secara garis besar, algoritma TLBO dapat dijabarkan dalam Gambar 5.

$$new\_X_i = X_i + rand(0,1) * (X_{teacher} - round[1 + rand(0,1)] * X_{mean}) \quad (11)$$

$$new\_X_i = \begin{cases} X_i + rand(0,1) * (X_i - X_j) & f(X_i) < f(X_j) \\ X_i + rand(0,1) * (X_j - X_i) & \text{otherwise} \end{cases} \quad (12)$$

di mana $f(X_i)$ dan $f(X_j)$ merupakan nilai objektif dari masing-masing X_i dan X_j.

```
1: Mendefinisikan problem, menentukan fungsi objektif, inisialisasi parameter, menentukan kriteria berhenti
2: Inisialisasi ekosistem berisikan organisme dengan nilai acak sesuai dengan rentang pencarian
3: Mengevaluasi masing-masing organisme untuk mendapatkan nilai objektif dan menetapkan organisme terbaik
4: while (kriteria berhenti belum terpenuhi) do /* Main optimization looping /*
5:      for setiap organisme i do
6:          /* awal dari mutualism phase */
7:          memilih organisme secara acak sebagai organisme j
8:          memodifikasi organisme i dan j melalui operasi matematis pada Persamaan (5) dan (6)
9:          mengevaluasi nilai objektif dari organisme i dan j setelah interaksi
10:         menerima hasil modifikasi organisme i dan j apabila nilai objektif yang dihasilkan lebih baik
11:         /* akhir dari mutualism phase */
11:         /* awal dari commensalism phase */
12:         meilih organisme j secara acak dari ekosistem
13:         memodifikasi organisme i melalui operasi matematis pada Persamaan (7)
14:         mengevaluasi nilai objektif dari organisme i setelah interaksi
15:         menerima hasil modifikasi organisme i apabila nilai objektif yang dihasilkan lebih baik
16:         /* akhir dari commensalism phase */
16:         /* awal dari parasitism phase */
17:         memilih organisme j secara acak dari ekosistem
18:         membentuk "Parasite_Vector" melalui operasi matematis pada Persamaan (8)
19:         mengevaluasi nilai objektif dari Parasite_Vector
20:         mengubah organisme j menjadi Parasite_Vector apabila nilai objektif yang dihasilkan lebih baik
21:         /* akhir dari parasitism phase */
21:         menetapkan organisme terbaik (Obest) dalam ekosistem
21:     end for
22: end while
23: Mendapatkan output solusi terakhir berupa organisme terbaik (Obest)
```

Gambar 2. *Pseudo-code* dari algoritma SOS

```
01: Mendefinisikan problem, menentukan fungsi objektif, inisialisasi parameter, menentukan kriteria berhenti
02: Inisialisasi populasi berisikan partikel dengan nilai acak sesuai dengan rentang pencarian
03: Mengevaluasi masing-masing partikel untuk mendapatkan nilai objektif, menetapkan pbest dan gbest inisial
04: while (kriteria berhenti belum terpenuhi) do /* Main optimization looping /*
05:     for setiap partikel i do
06:         update kecepatan vi sesuai Persamaan (9)
07:         update xi = xi+vi
08:         mengevaluasi nilai objektif dari partikel i (xi)
09:         update pbesti apabila nilai objektif yang dihasilkan lebih baik
10:     end for
11:     update gbest dengan mengambil nilai pbest yang paling minimum
12: end while
13: Mendapatkan output solusi terakhir berupa partikel terbaik (gbest)
```

Gambar 3. *Pseudo-code* dari algoritma PSO

```
01: Mendefinisikan problem, menentukan fungsi objektif, inisialisasi parameter, menentukan kriteria berhenti
02: Inisialisasi sumber makanan dengan nilai acak sesuai dengan rentang pencarian, t=0, trial=0
03: Mengevaluasi masing-masing sumber makanan untuk mendapatkan nilai objektif
04: while (kriteria berhenti belum terpenuhi) do /* Main optimization looping /*
05:     /* Awal dari employed bee phase */
06:     for setiap sumber makanan i do
07:         Employed bee pergi ke sumber makanan i dan mencari sumber lain disekitarnya via waggle dance
            sesuai Persamaan (10)
08:         if sumber makanan i tidak meningkat then
09:             trial_i=trial_i+1
10:         else
11:             trial_i=0
12:         end
13:     end for
14:     /* Akhir dari employed bee phase */
15:     /* Awal dari onlooker bee phase */
16:     while (t ≤ jumlah sumber makanan)
17:         Onlooker bee mengeksplorasi sumber makanan secara acak berdasarkan informasi dari waggle dance
            oleh employeed bee
18:         t=t+1
19:     end
20:     /* Awal dari scout bee phase */
21:     if max(trial_i) > limit then
22:         mengganti sumber makanan i dengan solusi baru yang dibentuk dengan variabel acak
23:     end
24:     /* Akhir dari scout bee phase */
25:     update sumber makanan terbaik
26: end while
27: Mendapatkan output solusi terakhir berupa sumber makanan terbaik
```

Gambar 4. *Pseudo-code* dari algoritma ABC

```
01: Mendefinisikan problem, menentukan fungsi objektif, inisialisasi parameter, menentukan kriteria berhenti
02: Inisialisasi populasi berisikan partikel dengan nilai acak sesuai dengan rentang pencarian
03: Mengevaluasi masing-masing partikel untuk mendapatkan nilai objektif
04: while (kriteria berhenti belum terpenuhi) do /* Main optimization looping /*
05:     for setiap murid i do
06:         /* Teacher phase */
07:         meilih murid j secara acak dari populasi
08:         update solusi terbaik dalam populasi (X_teacher)
09:         memodifikasi murid i melalui operator matematis sesuai Persamaan (11)
10:         mengevaluasi nilai objektif dari murid i setelah teacher phase
11:         menerima hasil modifikasi murid i apabila nilai objektif yang dihasilkan lebih baik
12:         /* Learner phase */
13:         memilih murid j secara acak dari populasi
14:         memodifikasi murid i melalui learner phase sesuai Persamaan (12)
15:         mengevaluasi nilai objektif dari murid i setelah learner phase
16:         menerima hasil modifikasi murid i apabila nilai objektif yang dihasilkan lebih baik
17:     end for
18: end while
19: Mendapatkan output solusi terakhir berupa X_teacher
```

Gambar 5. *Pseudo-code* dari algoritma TLBO

Tabel 1. Transformasi solusi berbasis continuous menjadi permutasi

Lokasi	1	2	3	4	5	6	7	8
Solusi fasilitas berbasis *continuous*	0.34	0.29	0.87	0.98	0.02	0.49	0.45	0.61
Solusi fasilitas berbasis permutasi	3	2	7	8	1	5	4	6

Proses Optimasi Penentuan Tata Letak Fasilitas dengan Menggunakan Algoritma Metaheuristik

Proses optimasi penentuan tata letak fasilitas dari masing-masing algoritma metaheuristik terbagi menjadi 4 tahapan yaitu:

Inisialisasi parameter

Penentuan fungsi objektif serta informasi pendukung kasus tata letak fasilitas seperti jarak antar lokasi dan frekuensi perpindahan pekerja antar fasilitas harus didefinisikan dengan jelas di awal. Selain itu, setiap algoritma metaheuristik memiliki parameter yang harus ditentukan sebelum proses pencarian dimulai.

Inisialisasi variabel solusi (populasi) secara acak dan evaluasi nilai objektif

Masing-masing dari algoritma metaheuristik akan melakukan inisialisasi variabel solusi awal secara acak. Adapun variabel solusi yang dibentuk berupa angka permutasi yang menyatakan hubungan antara penempatan fasilitas dan lokasi. Ketiga algoritma yang digunakan dalam penelitian ini bersifat *continuous*, namun, solusi yang harus dihasilkan berupa angka permutasi. Oleh karena itu, dibutuhkan adaptasi terhadap solusi yang dihasilkan oleh metaheuristik. Penelitian ini mengembangkan prosedur untuk mengubah representasi solusi dari nilai berbasis *continuous* menjadi nilai berbasis urutan atau permutasi. Representasi tata letak fasilitas dibuat sesuai dengan nilai indeks terhadap nilai continuous yang diurutkan menurut urutan naik. Representasi solusi ini digunakan untuk memecahkan masalah penentuan tata letak fasilitas. Contoh proses transformasi solusi yang semula dari berbasis *continuous* diubah menjadi berbasis permutasi dapat diilustrasikan melalui Tabel 1. Setelah proses transformasi selesai, variabel solusi yang menandakan hubungan penempatan fasilitas terhadap lokasi akan dievaluasi nilai objektifnya.

Menjalankan proses optimasi yang disimulasikan oleh algoritma metaheuristik

Proses optimasi akan dilakukan secara berulang-ulang dengan prinsip tertentu sesuai dengan karakteristik pola pencarian masing-masing algoritma. Simulasi pencarian pada PSO dilakukan berdasarkan konsep pergerakan partikel yang mengacu pada *pbest* dan *gbest*. Simulasi pencarian pada ABC akan melibatkan *employed bees*, *onlooker bees*, dan *scout bees* dan penerimaan solusi baru diijinkan apabila solusi berikutnya menjadi lebih optimal. Simulasi pencarian pada TLBO akan melalui *teacher phase* dan *learner phase* dan penerimaan solusi baru diijinkan apabila solusi berikutnya menjadi lebih optimal. Simulasi pencarian pada SOS akan melibatkan *mutualism phase*, *commensalism phase*, dan *parasitism phase* dimana penerimaan solusi baru diijinkan apabila solusi berikutnya menjadi lebih optimal pada setiap fase.

Kriteria berhenti

Informasi penempatan tata letak fasilitas terhadap lokasi yang dibawa oleh populasi akan dilanjutkan ke proses optimasi pada generasi selanjutnya sampai kriteria berhenti terpenuhi.

Hasil dan Pembahasan

Penelitian ini akan menginvestigasi kemampuan SOS dalam menentukan tata letak fasilitas proyek konstruksi akan diselidiki dengan menggunakan dua studi kasus, di antaranya: (1) kasus tata letak 11 fasilitas yang diadopsi dari Li and Love [10] dan (2) kasus tata letak fasilitas yang diambil dari proyek konstruksi bangunan tinggi di Surabaya. Sebagai perbandingan, akan dilibatkan algoritma-algoritma metaheuristik dasar yang lain seperti PSO, ABC, dan TLBO pada penelitian ini.

Sebanyak total 30 kali simulasi penyelesaian kasus optimasi akan dilakukan oleh setiap algoritma metaheuristik untuk menguji konsistensi. Total jarak tempuh pekerja terbaik yang didapatkan oleh setiap algoritma akan digunakan untuk mendapatkan nilai rata-rata, standar deviasi, terbaik dan terburuk. Parameter yang digunakan oleh setiap metode diatur pada Tabel 2. Masing-masing algoritma metaheuristik menggunakan jumlah populasi (*Npop*) dan jumlah iterasi maksimum (*MaxIt*) yang sama yaitu masing-masing 50 dan 25.

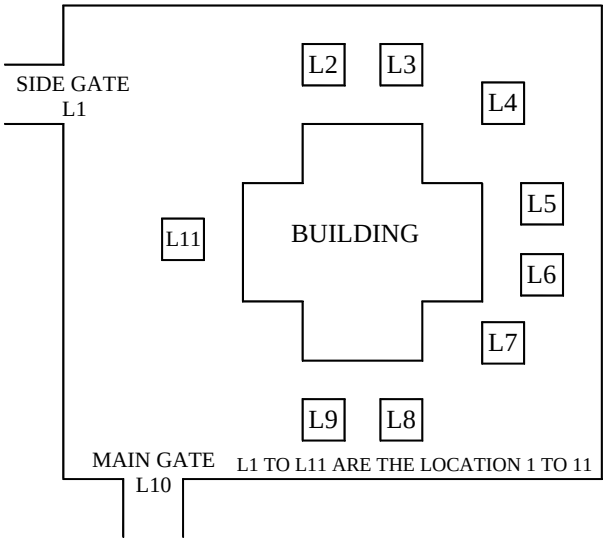
Studi Kasus 1: Problem Tata Letak 11 Fasilitas Proyek

Pada kasus yang di adopsi dari Li dan Love [10], terdapat sejumlah 11 fasilitas dan 11 lokasi, seperti yang terlihat pada Gambar 6. Ada dua buah fasilitas yang memiliki lokasi yang permanen. Sementara itu, frekuensi perpindahan harian dan jarak antar lokasi dapat dilihat pada Tabel 3 dan Tabel 4. Fasilitas-fasilitas yang terdapat pada studi kasus ini diantara lain: (1) *Site office* (SO). (2) *Falsework shop* (FS). (3) *Labor residence* (LR). (4) *Storeroom* 1 (S1). (5) *Storeroom* 2 (S2). (6) *Carpentry workshop* (CW). (7) *Reinforcement steel workshop* (RW). (8) *Side gate* (SG), lokasi permanen. (9) *Electrical, water, and utility control room* (UR). (10) *Concrete batch workshop* (BW). (11) *Main gate* (MG), lokasi permanen.

Tabel 2. Pengaturan parameter untuk setiap algoritma metaheuristik

PSO	ABC	TLBO	SOS
$c_1 = 2$ $c_2 = 2$ $w = 0.4\text{-}0.9$ $Npop = 50$ $MaxIt = 25$	$Limit = 100$ $Npop = 50$ $MaxIt = 25$	$Npop = 50$ $MaxIt = 25$	$Npop = 50$ $MaxIt = 25$

Tabel 5 mencatat hasil rata-rata, terbaik, terburuk, dan standar deviasi yang setelah simulasi dari masing-masing metode selesai dilakukan. Terlihat bahwa keempat algoritma dari masing-masing algoritma metaheuristik mampu menghasilkan nilai terbaik 12546 m. Namun, hanya SOS dan TLBO yang dapat menghasilkan nilai terbaik pada setiap perulangan simulasi ditunjukkan dari nilai rata-rata yang juga sama dengan nilai terbaik yaitu 12546 m, dan oleh karena itu standar deviasinya juga bernilai 0.



Gambar 6. Tata letak dari proyek pada studi kasus 1

Tabel 3. Matriks jarak antara lokasi pada studi kasus 1 (dalam satuan meter)

	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11
L1	-	15	25	33	40	42	47	55	35	34	20
L2		-	10	18	25	27	32	42	50	45	35
L3			-	8	15	17	22	32	52	55	45
L4				-	7	9	14	24	44	49	53
L5					-	2	7	17	37	42	52
L6						-	5	15	35	40	50
L7							-	10	30	35	40
L8								-	20	25	35
L9									-	5	15
L10										-	10
L11											-

Tabel 4. Matriks frekuensi pergerakan pekerja antar fasilitas pada studi kasus 1

	SO	FS	LR	S1	S2	CW	RW	SG	UR	BW	MG
SO	-	5	2	2	1	1	4	1	2	9	1
FS		-	2	5	1	2	7	8	2	3	8
LR			-	7	4	4	9	4	5	6	5
S1				-	8	7	8	1	8	5	1
S2					-	3	4	1	3	3	6
CW						-	5	8	4	7	5
RW							-	7	6	3	2
SG								-	9	4	8
UR									-	5	3
BW										-	5
MG											-

Tabel 5. Nilai rata-rata, terbaik, terburuk, dan standar deviasi setelah 30 kali simulasi (dalam meter)

Metode	Rata-rata	Terbaik	Terburuk	Standar deviasi
PSO	12588.0	12546	12678	68.35
ABC	12642.2	12546	12904	110.93
TLBO	12546.0	12546	12546	0
SOS	12546.0	12546	12546	0

Tabel 6. Penempatan fasilitas terhadap lokasi pada hasil usulan tata letak yang optimal

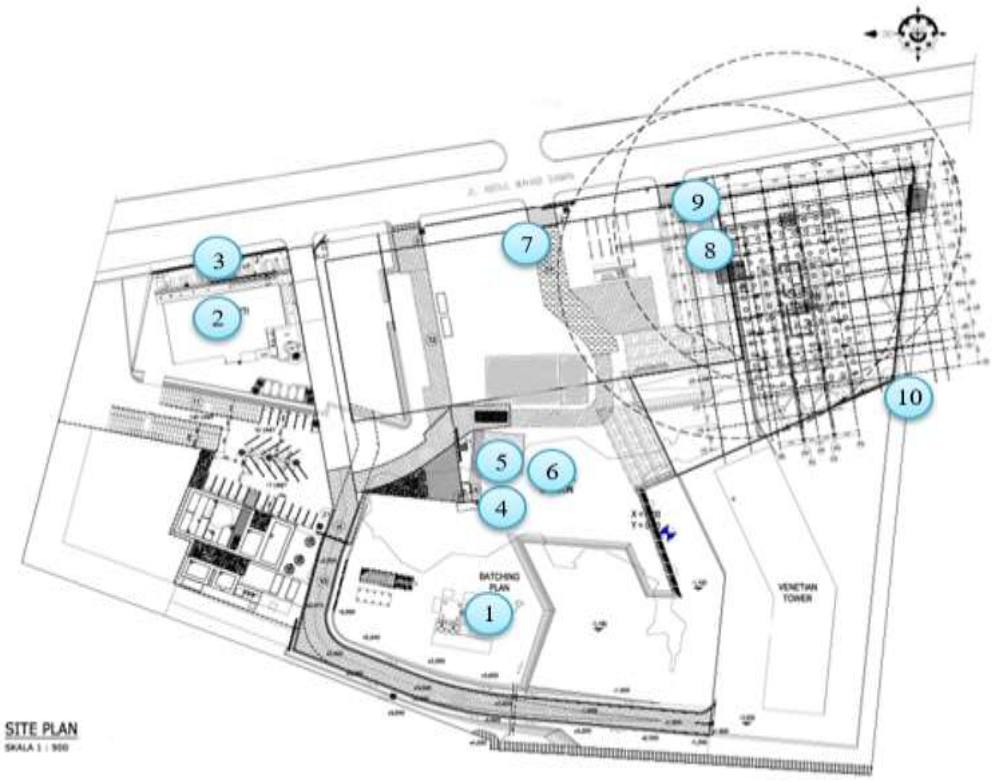
Lokasi	1	2	3	4	5	6
Tata letak optimal	SG	UR	RW	LR	CW	S1
Lokasi	7	8	9	10	11	Nilai objektive (meter)
Tata letak optimal	S2	BW	SO	MG	FS	39199.6

Algoritma dengan performa terburuk pada kasus ini adalah ABC, dengan nilai rata-rata, standar deviasi, serta nilai terburuk yang lebih tinggi jika dibandingkan ketiga metode lainnya. Tabel 6 menunjukkan penempatan fasilitas pada lokasi menurut tata letak optimal yang didapatkan dari simulasi menggunakan algoritma metaheuristik.

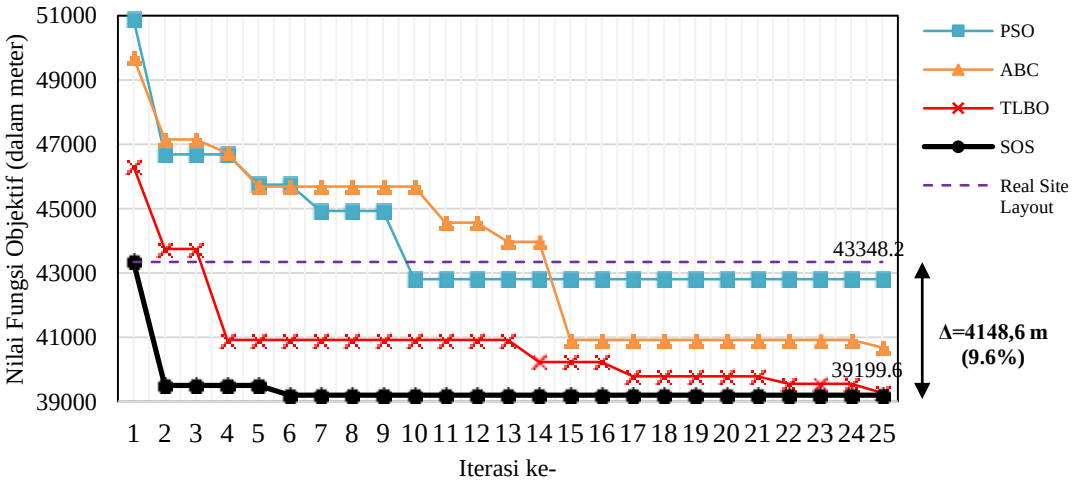
Studi Kasus 2: Problem Tata Letak Fasilitas Proyek Bangunan Bertingkat di Surabaya

Penelitian ini mengadopsi denah tata letak salah satu proyek konstruksi bangunan bertingkat yang ada di Surabaya. Berdasarkan data fasilitas pada proyek tersebut, terdapat total 32 jenis fasilitas mula-mula. Namun, setelah dilakukan observasi lapangan secara langsung, tidak semua fasilitas sementara tersebut aktif digunakan. Di samping itu, ada beberapa fasilitas yang tidak mengalami pergerakan dan tidak berhubungan dengan proses pelaksanaan yang pada saat dilakukan observasi. Dalam penelitian ini, dilakukan pengeliminasian pada fasilitas-fasilitas sementara yang tidak aktif yang menyebabkan total fasilitas sementara berkurang menjadi 10 jenis yang akan digunakan untuk proses optimasi. Lokasi untuk penempatan fasilitas dapat dilihat pada Gambar 7. Fasilitas-fasilitas tersebut antara lain: (1) *Batching plant* (BP). (2) Direksi keet (DK). (3) *Workshop bekisting* (WB). (4) *Gate* masuk (GM), ditempatkan permanen pada lokasi 4. (5) Pos jaga (PJ), ditempatkan permanen pada lokasi 5. (6) Fabrikasi GRC (FG). (7) Kantor kontraktor (KI). (8) Penyimpanan besi (SB). (9) Fabrikasi besi 1 (FB1). (10) Fabrikasi besi 2 (FB2).

Sementara itu, frekuensi perpindahan harian dan jarak antar lokasi dapat dilihat pada Tabel 7 dan Tabel 8. Jarak antara fasilitas sementara yang akan ditentukan bersifat *reversible*. Sebagai contoh, jarak antara batching plant dan direksi keet akan sama dengan jarak direksi keet ke batching plant.



Gambar 7. Tata letak dari proyek pada studi kasus 2



Gambar 8. Grafik perbandingan karakteristik konvergensi untuk setiap algoritma metaheuristik

Tabel 7. Matriks jarak antara lokasi pada studi kasus 2 (dalam satuan meter)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	-	138.6	156.2	32.6	39.2	49.4	138.1	170.2	174.3	150.3
2		-	19.1	106.1	100.1	111.8	128.3	160.4	164.5	188.2
3			-	125.1	119.2	130.8	111.7	143.8	147.9	207.3
4				-	12.3	22.6	111.3	143.3	147.4	123.4
5					-	11.6	98.9	131	135.1	111.1
6						-	88.7	120.7	124.8	100.7
7							-	32	36.2	104.1
8								-	9.3	41.9
9									-	102.1
10										-

Tabel 8. Matriks frekuensi pergerakan pekerja antar fasilitas pada studi kasus 2

	BP	DK	WB	GM	PJ	FG	KI	SB	FB1	FB2
BP	-	10	8	9	3	9	0	0	0	0
DK		-	8	12	8	9	11	5	0	1
WB			-	4	3	8	0	0	0	0
GM				-	6	15	10	10	8	5
PJ					-	9	5	3	2	1
FG						-	0	0	0	0
KI							-	7	7	10
SB								-	25	27
FB1									-	16
FB2										-

Tabel 9. Nilai rata-rata, terbaik, terburuk, dan standar deviasi setelah 30 kali simulasi (dalam meter)

Metode	Rata-rata	Terbaik	Terburuk	Standar deviasi
PSO	41280.4	39199.6	46178.0	2484.8
ABC	40673.0	39199.6	45828.6	1836.4
TLBO	39302.3	39199.6	40235.0	236.8
SOS	39199.6	39199.6	39199.6	0

Tabel 10. Perbandingan penempatan fasilitas terhadap lokasi proyek semula dan setelah dioptimasi oleh SOS

Lokasi	1	2	3	4	5	6	7	8	9	10	Nilai objektif (meter)
Tata letak semula	BP	DK	WB	GM	PJ	FG	KI	SB	FB1	FB2	43348.2
Tata letak setelah dioptimasi oleh SOS	FG	BP	WB	GM	PJ	DK	SB	FB2	FB1	KI	39199.6

Tabel 9 mencatat hasil rata-rata, terbaik, terburuk, dan standar deviasi yang setelah simulasi dari masing-masing metode selesai dilakukan. Terlihat bahwa keempat algoritma dari masing-masing algoritma metaheuristik mampu menghasilkan nilai terbaik 39199.6 m. Namun, hanya SOS yang dapat menghasilkan nilai terbaik pada setiap perulangan simulasi ditunjukkan dari nilai rata-rata yang juga sama dengan nilai terbaik yaitu 39199.6 m, dan oleh karena itu standar deviasinya juga bernilai 0. Algoritma dengan performa terburuk pada kasus ini adalah PSO, dengan nilai rata-rata, standar deviasi, serta nilai terburuk yang lebih tinggi jika disbandingkan ketiga metode lainnya. Disamping itu, SOS menghasilkan nilai objektif yang lebih kecil 4148.6 meter dibandingkan dengan nilai objektif dari tata letak fasilitas proyek semula seperti yang terlihat pada Tabel 10. Hal ini membuat algoritma SOS dapat menghasilkan alternatif solusi 9.6% lebih baik dibandingkan dengan tata letak fasilitas proyek semula.

Selain itu, analisa karakteristik konvergensi dari masing-masing metode metaheuristik dilakukan guna mempelajari seberapa cepat suatu metode dalam menemukan solusi optimal. Gambar 8 menunjukkan perbandingan perubahan nilai objektif pada setiap iterasi, dari iterasi pertama hingga iterasi terakhir, untuk masing-masing metode. Terlihat pada simulasi tersebut, SOS mampu menemukan solusi optimal pada iterasi ke-7. Dapat disimpulkan bahwa metode SOS mampu mencapai titik konvergen lebih cepat dan lebih optimal jika dibandingkan dengan ketiga metode lainnya.

Simpulan

Penelitian ini mengimplementasikan SOS dalam penentuan tata letak fasilitas yang didasarkan kepada jarak tempuh pekerja proyek konstruksi. Dua buah studi kasus optimasi penentuan tata letak fasilitas proyek diadopsi untuk menguji performa dari SOS, diantaranya: problem tata letak 11 fasilitas proyek dan problem tata letak fasilitas proyek bangunan bertingkat di Surabaya. Dalam kesempatan ini, PSO, ABC, dan TLBO ditunjuk sebagai pembanding. Analisa hasil menunjukkan bahwa SOS memiliki akurasi pencarian solusi yang tinggi pada setiap kasus bila dibandingkan dengan algoritma lainnya. Hal ini terbukti dari solusi yang dihasilkan oleh SOS selalu menjadi yang paling optimal. Di sisi lain, SOS mampu secara konsisten menghasilkan solusi optimal bila dibandingkan dengan algoritma lainnya. SOS mampu menghasilkan nilai rata-rata dan standar deviasi terkecil di antara algoritma-algoritma lainnya dari hasil 30 kali perulangan pada masing-masing kasus. Algoritma metaheuristik yang baik biasanya memiliki kemampuan mencari area pencarian global baru yang berpotensi memiliki solusi optimal (dikenal dengan kemampuan eksplorasi) serta kemampuan menyelidiki solusi lokal yang terbaik dalam area pencarian tertentu (dikenal dengan kemampuan eksploitasi). Keseimbangan skema pencarian SOS yang terbagi menjadi kemampuan eksplorasi (*mutualism phase* dan *commensalism phase*) serta kemampuan eksploitasi (*parasitism phase*) mampu membuat SOS unggul dibanding metode metaheuristik lainnya. Sebagai tambahan, melalui *parasitism phase*, SOS juga mampu untuk menghilangkan solusi inferior. Hal ini menjadikan SOS sebagai salah satu alternatif metode yang cukup potensial untuk menghadapi berbagai permasalahan di bidang tata letak fasilitas proyek.

Daftar Pustaka

1. Adrian, A. M., Utamima, A., and Wang, K.-J., A Comparative Study of GA, PSO and ACO for Solving Construction Site Layout Optimization, *KSCE Journal of Civil Engineering*, 19(3), 2015, pp. 520-527.
2. Tommelein, I., Levitt, R., and Hayes-Roth, B., SightPlan Model for Site Layout, *Journal of Construction Engineering and Management*, 118(4), 1992, pp. 749-766.

3. Simmons, D. M., Single Row Space Allocation: An Ordering Algorithm, *Operations Research*, 17(5), 1969, pp. 812–826.
4. Picard, J.-C. and Queyranne, M., On the One-Dimensional Space Allocation Problem, *Operations Research*, 29(2), 1981, pp. 371-391.
5. Love, R. F. and Wong, J. Y., On Solving a Single Row Space Allocation Problem with Integer Programming, *INFOR: Information Systems and Operational Research*, 14(2), 1976, pp. 139-143.
6. Ravi Kumar, K., Hadjinicola, G. C., and Lin, T.-l., A Heuristic Procedure for the Single-Row Facility Layout Problem, *European Journal of Operational Research*, 87(1), 1995, pp. 65-73.
7. Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
8. Kennedy, J. and Eberhart, R., Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.
9. Dorigo, M., Maniezzo, V., and Colorni, A., Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1), 1996, pp. 29-41.
10. Li, H. and Love, P. E. D., Site-Level Facilities Layout Using Genetic Algorithms, *Journal of Computing in Civil Engineering*, 12(4), 1998, pp. 227-231.
11. Zouein, P., Harmanani, H., and Hajar, A., Genetic Algorithm for Solving Site Layout Problem with Unequal-Size and Constrained Facilities, *Journal of Computing in Civil Engineering*, 16(2), 2002, pp. 143-151.
12. Lee, K.-Y., Roh, M.-I., and Jeong, H.-S., An Improved Genetic Algorithm for Multi-Floor Facility Layout Problems Having Inner Structure Walls and Passages, *Computers & Operations Research*, 32(4), 2005, pp. 879-899.
13. Önüt, S., Tuzkaya, U. R., and Doğaç, B., A Particle Swarm Optimization Algorithm for the Multiple-Level Warehouse Layout Design Problem, *Computers & Industrial Engineering*, 54(4), 2008, pp. 783-799.
14. Zhang, H. and Wang, J. Y., Particle Swarm Optimization for Construction Site Unequal-Area Layout, *Journal of Construction Engineering and Management*, 134(9), 2008, pp. 739-748.
15. Lam, K. C., Ning, X., and Ng, T., The Application of the Ant Colony Optimization Algorithm to the Construction Site Layout Planning Problem, *Construction Management and Economics*, 25(4), 2007, pp. 359-374.
16. Liang, L. Y. and Chao, W. C., The Strategies of Tabu Search Technique for Facility Layout Optimization, *Automation in Construction*, 17(6), 2008, pp. 657-669.
17. Yahya, M. and Saka, M. P., Construction Site Layout Planning using Multi-objective Artificial Bee Colony Algorithm with Levy flights, *Automation in Construction*, 38, 2014, pp. 14-29.
18. Setiawan, I. L. and Palit, H. C., Perbandingan Kombinasi Genetic Algorithm–Simulated Annealing dengan Particle Swarm Optimization pada Permasalahan Tata Letak Fasilitas, *Jurnal Teknik Industri*, 12(2), 2010, pp. pp. 119-124.
19. Mak, K. L., Wong, Y. S., and Chan, F. T. S., A Genetic Algorithm for Facility Layout Problems, *Computer Integrated Manufacturing Systems*, 11(1–2), 1998, pp. 113-127.
20. Prayogo, D., An Innovative Parameter-Free Symbiotic Organisms Search (SOS) for Solving Construction-Engineering Problems, 2015, PhD Thesis, Department of Construction Engineering, National Taiwan University of Science and Technology.
21. Cheng, M.-Y., Firdausi, P. M., and Prayogo, D., High-performance concrete compressive strength prediction using Genetic Weighted Pyramid Operation Tree (GWPOT), *Engineering Applications of Artificial Intelligence*, 29, 2014, pp. 104-113.
22. Cheng, M.-Y., Prayogo, D., and Tran, D.-H., Optimizing Multiple-Resources Leveling in Multiple Projects Using Discrete Symbiotic Organisms Search, *Journal of Computing in Civil Engineering*, 30(3), 2016, pp. 04015036.
23. Prayogo, D., Cheng, M.-Y., and Prayogo, H., A Novel Implementation of Nature-inspired Optimization for Civil Engineering: A Comparative Study of Symbiotic Organisms Search, *Civil Engineering Dimension*, 19(1), 2017, pp. 36-43.
24. Tran, D.-H., Cheng, M.-Y., and Prayogo, D., A novel Multiple Objective Symbiotic Organisms Search (MOSOS) for time–cost–labor utilization tradeoff problem, *Knowledge-Based Systems*, 94, 2016, pp. 132-145.
25. Cheng, M.-Y., Chiu, C.-K., Chiu, Y.-F., Wu, Y.-W., Syu, Z.-L., Prayogo, D., and Lin, C.-H., SOS Optimization Model for Bridge Life Cycle Risk Evaluation and Maintenance Strategies, *Journal of the Chinese Institute of Civil and Hydraulic Engineering*, 26(4), 2014, pp. 293-308.
26. Cheng, M.-Y. and Prayogo, D., Modeling the Permanent Deformation Behavior of Asphalt Mixtures using a Novel Hybrid Computational Intelligence, *ISARC 2016 - 33rd International Symposium on Automation and Robotics in Construction*, Auburn, USA, 2016, pp. 1009-1015. International Association for Automation and Robotics in Construction.
27. Koopmans, T. and Beckmann, M., Assignment Problems and the Location of Economic Activities, *Econometrica*, 25(1), 1957, pp. 53-76.

28. Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P., and Querido, T., A survey for the Quadratic Assignment Problem, *European Journal of Operational Research*, 176(2), 2007, pp. 657-690.
29. Karaboga, D. and Basturk, B., A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm, *Journal of Global Optimization*, 39(3), 2007, pp. 459-471.
30. Rao, R. V., Savsani, V. J., and Vakharia, D. P., Teaching–learning-based Optimization: A Novel Method for Constrained Mechanical Design Optimization Problems, *Computer-Aided Design*, 43(3), 2011, pp. 303-315.