Expert Systems with Applications 39 (2012) 9642–9650

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A hybrid swarm intelligence based particle-bee algorithm for construction site layout optimization

Li-Chuan Lien [a,*], Min-Yuan Cheng [b]

[a] National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei 106, Taiwan, ROC
[b] Department of Construction Engineering, National Taiwan University of Science and Technology, #43, Sec. 4, Keelung Rd., Taipei 106, Taiwan, ROC

ARTICLE INFO

Keywords:
Construction site layout
Swarm intelligence
Bee algorithm
Particle swarm optimization
Particle-bee algorithm

ABSTRACT

The construction site layout (CSL) design presents a particularly interesting area of study because of its relatively high level of attention to usability qualities, in addition to common engineering objectives such as cost and performance. However, it is difficult combinatorial optimization problem for engineers. Swarm intelligence (SI) was very popular and widely used in many complex optimization problems which was collective behavior of social systems such as honey bees (bee algorithm, BA) and birds (particle swarm optimization, PSO). This study proposed an optimization hybrid swarm algorithm namely particle-bee algorithm (PBA) based on a particular intelligent behavior of honey bee and bird swarms by integrates theirs advantages. This study compares the performance of PBA with that of BA and PSO for hypothetical construction engineering of CSL problems. The results show that the performance of PBA is comparable to those of the mentioned algorithms and can be efficiently employed to solve those hypothetical CSL problems with high dimensionality.

1. Introduction

Construction site layout (CSL) problems are particularly interesting because in addition to common engineering objectives such as cost and performance, facility design is especially concerned with aesthetics and usability qualities of a layout (Michalek, Choudhary, & Papalambros, 2002). The CSL problem identifies a feasible location for a set of interrelated objects that meet all design requirements and maximizes design quality in terms of design preferences while minimizing total cost associated with interactions between these facilities. Pairwise costs usually reflect transportation costs and/or inter-facility adjacency preferences (Anjos & Vannelli, 2002; Michalek et al., 2002). CSL problems arise in the design of hospitals, service centers and other facilities (Yeh, 2006). However, all such problems are known as "NP-hard" and because of the combinatorial complexity, it cannot be solved exhaustively for reasonably sized layout problems (Yeh, 2006). For n facilities, the number of possible alternatives is $n!$, which gives large numbers even for small n values. When 10 facilities are involved, possible alternatives number well over 3,628,800, and 15 facilities have possible alternatives numbering in the billions. In practical application, however, a project with $n = 15$ is still considered small (Yeh, 2006).

In the past, knowledge and artificial intelligence based methods have been applied to solving CSL problems, e.g., knowledge based systems have been developed to provide users with problem-specific heuristic knowledge to facilitate appropriate facility allocations (Cheng, 1992; Tommelein, Levitt, & Confrey, 1991). For the AI-based algorithms, Elbeitagi and Hegazy (2001) used a hybrid neural network to identify optimal site layout. Yeh (2006) applied annealed neural networks to solve construction site-level CSL problems. Other well-known algorithms, e.g., tabu search (TS), simulated annealing (SA) and genetic algorithms (GAs) are used widely to solve site layout problems. TS is a local search method, which is used for the laying out of multi-floor facilities (Abdinnour-Helm & Hadley, 2000). SA is a method for solving combination problems generally applied to the layout design of multi-objective facilities (Suresh & Sahu, 1993). Gero and Kazakov (1997) incorporated the concept of genetic engineering into the GA system for solving building space layout problems. Li and Love (2000) and Osman, Georgy, and Ibrahim (2003) used GA to solve site layout problems in unequally sized facilities. The objective functions of the above-mentioned algorithms were to optimize the interaction between facilities, such as total inter-facility transportation costs and frequency of inter-facility trips. Hegazy and Elbeltagl (1999) developed a comprehensive system for site layout planning based on GA. Elbeitagi, Hegazy, Hosny, and Eldosouky (2001) presented a practical model for schedule-dependent site layout planning in construction by combining a knowledge-based system, fuzzy logic and GA. Those previous research focused on

* Corresponding author. Tel.: +886 2 27330004.
E-mail addresses: lclien@gmail.com (L.-C. Lien), myc@mail.ntust.edu.tw (M.-Y. Cheng).

0957-4174/$ - see front matter Crown Copyright © 2012 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.02.134

solving different optimization problems by applying those algorithms under different constraints which quality of solutions were limited by the capability of the algorithms.

Swarm intelligence (SI) has been of increasing interest to research scientists in recent years. SI was defined by Bonabeau et al. as any attempt to design algorithms or distributed problem-solving devices based on the collective behavior inspired of social insect colonies or other animals (Bonabeau, Dorigo, & Theraulaz, 1999). Bonabeau et al. focused primarily on the social behavior inspired of ants (Dorigo, 1992), fish (Li, 2003), birds (Kennedy & Eberhart, 1995), bees (Pham et al., 2006), etc. However, the term "swarm" can be applied more generally to refer to any restrained collection of interacting agents or individuals. Although bees swarming around a hive is the classical example of "swarm", swarms can easily be extended to other systems with similar architectures.

A few models have been developed to model the intelligent behaviors of honeybee swarms and applied to solve combinatorial type problems. Yang (2005) presented a virtual bee algorithm (VBA) that is effective when applied to function optimization problems. VBA was tested on two functions with two parameters, single-peaked and multi-peaked, respectively. Results show the VBA as significantly more efficient than GA. Karaboga and Akay (2009) presented an artificial bee colony (ABC) algorithm and expanded its experimental results (Basturk & Karaboga, 2006). It has been pointed out that the ABC algorithm outperforms GA for functions exhibiting multi-modality or uni-modality. Pham et al. (2006) presented an original bee algorithm (BA) and applied to two standard functional optimization problems with two and six dimensions. Results demonstrated BA able to find solutions very close to the optimum, showing that BA generally outperformed GA. Ozbakir, Baykasog, and Tapkan (2010) developed a modified BA (Pham et al., 2006) to solve generalized assignment problems (GAP) that presented an ejection chain neighborhood mechanism. Results found that the proposed BA offers the potential to solve GAP. However, while BA (Pham et al., 2006) offers the potential to conduct global searches and uses a simpler mechanism in comparison with GA, it dependence on random search makes it relatively weak in local search activities and does not records past searching experiences during the optimization search process. For instance, a flock of birds may be thought of as a swarm whose individual agents are birds. Particle swarm optimization (PSO), which has become quite popular for many researchers recently (Parsopoulos & Vrahatis, 2007; Tsai, 2010), models the social behavior inspired of birds (Pham et al., 2006). PSO potentially used in local searching, and records past searching experiences during optimization search process. However, it converges early in highly discrete problems and traps into the local optimum solution (Korenaga, Hatanaka, & Uosaki, 2006).

Hence, in order to improve BA and PSO, this study proposed an improved optimization hybrid swarm algorithm called the particle-bee algorithm (PBA) that imitates a particular intelligent behavior inspired of bird and honey bee swarms and integrates their advantages. In addition, this study also proposed a neighborhood-windows (NW) technique for improving PBA search efficiency and proposed a self-parameter-updating (SPU) technique for preventing trapping into a local optimum in high dimensional problems. This study compares the performance of PBA algorithm with that of BA (Pham et al., 2006) and PSO for hypothetical construction engineering of CSL problems.

2. Hybrid swarm algorithm particle-bee algorithm

2.1. Bee algorithm (BA)

Bee algorithm (BA) is an optimization algorithm inspired by the natural foraging behavior of honeybees (Eberhart, Shi, & Kennedy, 2001). BA flowchart is shown in Fig. 1. BA (Pham et al., 2006) requires the setting of a number of parameters, including number of scout bees (n), number of elite sites selected from n visited sites (e), number of best sites out of n visited sites (b), number of bees recruited for elite e sites (n_1), number of bees recruited for best b sites (n_2), number of bees recruited for other visited sites (r), and neighborhood (ngh) of bees dance search and stopping criterion.

Step (1) Initialize scout bees
BA starts with n scout bees placed randomly in the search space.

Step (2) Evaluate fitness
Start the loop and evaluate scout bee fitness.

Step (3) Select elite sites (e) from scout bees
Scout bees that have the highest fitness are chosen as elite bees, and sites they visit are chosen for neighborhood search.

Step (4) Recruit bees (n_1) start neighborhood dance search
The algorithm conducts searches in the neighborhood of selected sites, assigning more recruit bees to dance near to elite sites. Recruit bees can be chosen directly according to the fitness associated with their specific dance sites Eq. (1).

$$x_{id}(t+1) = x_{id}(t) \times (Rand - 0.5) \times 2 + x_{id}(t) \quad (1)$$

where x_i is ith x and i = 1 to n; d is dimension in x_i and d = 1 to D, t is iteration; $x_{id}(t+1)$ is dth dimension in ith x and in $t+1$ iteration; $x_{id}(t)$ is dth dimension in ith x and in t iteration; $Rand$ is a uniformly distributed real random number within the range 0 to 1; n is number of scout bees.

Step (5) Select best sites (b) from scout bees
Otherwise, scouts bees with the secondary highest fitness are chosen as best bees, and sites they visit are chosen for neighborhood search.

Step (6) Recruit bees (n_2) start neighborhood dancing search
The algorithm conducts searches in the neighborhood of the selected sites, assigning more recruit bees to dance near the best sites. Recruit bees can be chosen directly according to the fitness associated with dancing sites Eq. (1).
Elite bees differ from best bees as the former focus on local search in order to search the local optimum solution, and the latter focus on global search in order to avoid missing other potential global optimum solutions. Alternatively, fitness

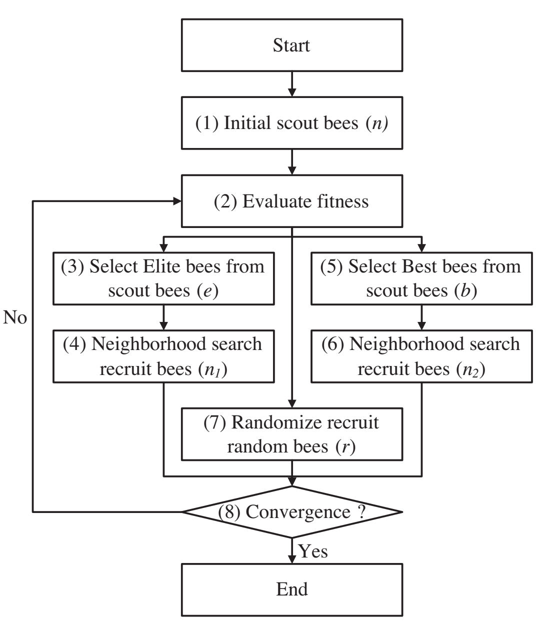

Fig. 1. Bee algorithm flowchart.