Applied Soft Computing 46 (2016) 220–229

Contents lists available at ScienceDirect

Applied Soft Computing

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

Hybridizations of genetic algorithms and neighborhood search metaheuristics for fuzzy bus terminal location problems

Saman Babaie-Kafaki [a], Reza Ghanbari [b], Nezam Mahdavi-Amiri [c,*]

[a] Department of Mathematics, Faculty of Mathematics, Statistics and Computer Sciences, Semnan University, Semnan, Iran
[b] Department of Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran
[c] Faculty of Mathematical Sciences, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 1 October 2012
Received in revised form 1 December 2015
Accepted 7 March 2016
Available online 5 May 2016

Keywords:
Hybrid algorithm
Memetic algorithm
Genetic algorithm
Simulated annealing
Variable neighborhood search
Fuzzy facility location problem
Quadratic assignment problem

ABSTRACT

We propose modified hybridizations of genetic algorithms with some neighborhood search based metaheuristics. In our hybrid algorithms, we consider gradually increasing probability for the application of the neighborhood search procedure on the best individuals as the number of iterations of the genetic algorithm increases. We implement the proposed hybrid algorithms and compare their performance with two other recently proposed hybrid algorithms which, in contrast, use the neighborhood search procedure on all the individuals of the population, two hybrid algorithms applying simulated annealing on the best individual in the papulation in every iteration and three non-hybrid metaheuristic algorithms. To investigate the effectiveness of the proposed algorithms, we apply the algorithms to our proposed fuzzy bus terminal location problem models. The fuzzy model is considered to have fuzzy number of passengers corresponding to the nodes as well as fuzzy neighborhoods, together with preassigned lower and upper bounds for the number of required terminals. The algorithms are tested on a variety of randomly generated large scale fuzzy bus terminal location problems in with fuzzy cost coefficients. The fuzzy objective is transformed into a crisp one by use of a ranking function. The computational experiments demonstrate the effectiveness of the proposed algorithms on large scale problems. Finally, to show the effectiveness of our proposed hybridizations, we also make a comparative study of our algorithms on crisp facility location and quadratic assignment test problems.

1. Introduction

The continuing rise in complexity of systems has lead to new model developments demanding effective solution strategies. In this respect, fuzziness in data and application of soft computing methodologies such as evolutionary algorithms have attracted extensive interests. Metaheuristic and exact methods have both been used to solve a vast variety of practical decision making problems. Although exact methods are useful tools, they are not usually efficient enough for solving large scale problems. Thus, researchers often pay attention to metaheuristic methods to encounter large scale problems. Some of the most well known metaheuristic algorithms are genetic algorithm (GA) [19], simulated annealing (SA) [23], variable neighborhood search (VNS) [20], ant colony optimization (ACO) [8] and tabu search (TS) [18]. To increase the efficiency of the GA, researchers made some efforts to hybridize GA with some neighborhood search based procedures such as SA, VNS, local search, TS and ant colony algorithm. Most researchers consider the neighborhood search algorithm as an operator of the GA. Sample algorithms include hybridization of GA and SA studied by Han et al. [21] for solving nonlinear channel blind equalization, Leung et al. [35] for solving two dimensional orthogonal packing problem, Tanga et al. [48] for the construction of near-Moore digraphs and Wang and Zheng [51] for solving job-shop scheduling problems, hybridization of GA and VNS studied by Gao et al. [12] for solving flexible job shop scheduling problems and Tavakkoli-Moghaddam et al. [50] for solving the flexible flow line scheduling problem with processor blocking, hybridization of GA and TS studied by Drezner [10] and Taillard [49] for solving quadratic assignment problem; hybridization of GA and local search studied by Jaszkiewicz and Kominek [25] for solving a vehicle routing problem, and hybridization of GA and ant colony algorithm studied by Lee et al. [34] for solving multiple sequence alignment problem. In addition to the above hybrid algorithms,

* Corresponding author at: Faculty of Mathematical Sciences, Sharif University of Technology, P.O. Box:11365-9415, Tehran, Iran.
E-mail addresses: sbk@semnan.ac.ir (S. Babaie-Kafaki), rghanbari@um.ac.ir (R. Ghanbari), nezamm@sharif.edu (N. Mahdavi-Amiri).

http://dx.doi.org/10.1016/j.asoc.2016.03.005
1568-4946/

other approaches have also been proposed. In the hybrid algorithm proposed by Alba et al. [2], the obtained solutions from application of GA are sent to a neighborhood search procedure for possible improvement. Chiu et al. [7] coupled GA with SA in an iterative manner, that is, after a number of applications of GA, SA is executed on each solution obtained by GA and the best solutions identified by SA are sent to the GA to be considered for the next generations. This coupled procedure continues until a stopping condition is satisfied.

Although several algorithms have been proposed to apply the neighborhood search based algorithms on all the individuals of the population (see [10,12,21,25,35,48–51]), but application of the neighborhood search algorithm on a part of population have also been considered; sample works include the hybridization of GA and TS proposed by Kit et al. [27] for solving the vehicle routing problem with time windows, in which TS is applied to an increasing percentage of the population, and the hybridization of GA and a local search procedure proposed by Pasia et al. [42] for solving a bi-objective permutation flowshop problem, that is based on a path relinking approach.

Another hybridization of GA and SA was used by Hong et al.[24] and Lin et al. [36]. In [36], a hybridization of GA and SA was proposed appling SA on the best individual of the pervious population. But, in [24], SA starts from the best individual of the temporary population before the next generation. The main difference between the proposed algorithm in [24] and the one in [36] is in the order of using SA on the best individual. In other words, in [24], SA starts when the generation of the temporary population finishes. But, in [36], the generation of the temporary population starts when SA is terminated. The hybridization of a genetic algorithm and a local search is also very popular in soft computing. Proposed algorithms based on this hybridization are named as memetic algorithms or genetic local search algorithms; for examples, see [25,30]. Several design issues related to, “when, where and how often to apply the local search in a memetic algorithm” are studied in [29,45]. We refer to [44] for further investigations.

Here, we consider a probability for application of the neighborhood search procedure on the best individual of the new population. The probability is increased as the number of iterations of GA increases.

From the modeling perspective, in real applications, the strong crisp assumptions for a facility location problem, like exact values for demands and distances, are seldom realistic; see [31]. To make the problem more realistic, we use uncertainty in the input data. A popular approach to consider uncertain data makes use of fuzzy data. Here, we state a formulation of an special facility location problem, namely fuzzy bus terminal location problem. In contrast to the crisp formulation proposed in [16], for the fuzzy formulation, the number of passengers corresponding to the nodes and the neighborhoods are considered to be fuzzy. Also, to be practical, we impose lower and upper bounds on the number of required terminals. To show the efficiency of the our hybrid algorithms, we use these algorithms on a real world public transportation problem, namely fuzzy bus terminal location problem (FBTLP), an special case of the fuzzy facility location problem; see [39].

The remainder of our work is organized as follows. In Section 2, after a brief review of GA and SA, we present our approach for hybridization of these algorithms. Similarly, in Section 3, after a brief review of VNS, we present our approach for hybridization of GA and VNS. The fuzzy bus terminal location problem is discussed in Section 4. In Section 5, we test the implementations of our proposed algorithms on some large scale test problems and demonstrate the efficiency of the algorithms by numerical experiments. Finally, we conclude in Section 6.

2. A hybridization of genetic and simulated annealing algorithms

As pointed out in [4,19], genetic algorithms are members of a wider family of algorithms, named evolutionary algorithms. An evolutionary algorithm (EA) is a nondeterministic stochastic search/optimization method utilizing the notion of evolution. Each evolutionary algorithm needs a goal oriented selection operator in order to drive the search into promising regions of the solution space. This gives a direction to the evolution, by providing more chances of reproduction for high fitness value (objective function value) individuals. Selection alone cannot introduce any new individual into the population. New individuals are generated by the genetic operators, of which the most well known are crossover (or recombination) and mutation. The performance of an EA is mainly influenced by the genetic operators. A crossover operator takes two individuals and combines them (with a random process) to produce a new individual (or two individuals). A mutation operator is usually applied to each offspring after a crossover, by randomly altering each gene with a small probability. Mutation provides a small amount of random search, and helps to ensure that every point in the solution space has a nonzero probability of being examined.

As mentioned in [26] and [43], simulated annealing (SA) is analogous to the annealing process used in metallurgy, where a metal object is heated to near its melting temperature and then cooled slowly. SA is a local search algorithm capable of escaping from local optimum [43]. SA's ease of implementation, convergence properties and use of hill-climbing moves to escape local optima have made it a popular technique over the past two decades [1]. SA works through searching the set of all possible solutions, reducing the chance of being stuck in a poor local optimum by allowing moves to inferior solutions under the control of a randomized scheme [43]. SA has shown to be very efficient in practice (see [23,26,43]) and well developed in theory [1].

GA and SA are naturally motivated, general purpose approaches, representing global combinatorial optimization methods with complementary strengths and weaknesses. The main weakness of a genetic algorithm is its excessive computing requirements due to the need for searching a large number of points [33]. Also, a major weakness of a simulated annealing algorithm is that the randomness in generating a new trial point does not utilize the information gained during the search and therefore, the search may not easily be guided to more promising regions [46]. So, it may be a good idea to apply SA on promising regions of the solution space identified by GA. Recently, some hybridizations of GA and SA have been proposed (see [2,7,51]).

In several hybridization of GA and SA (e.g., [51]), SA is applied to all the individuals in the population. We name this approach of hybridization as HGASA.

Usually, in the early iterations of HGASA, GA may not find the significant parts of the feasible solution space. Therefore, implementation of SA on the members of available population may increase the running time without achieving any considerable improvement in the fitness of the population.

To circumvent this problem, we propose the following strategy. First, we consider a probability P_{SA} for applying SA in Step 4 of Algorithm 1. In the early iterations of our hybrid algorithm, it is more likely for the genetic operators to generate offsprings than the SA. Then, as we find the promising regions of the feasible solution space, we gradually increase the probability of applying SA on the members of the population. To increase P_{SA} in the hybrid algorithm, we use the following updating rule:

$$P_{SA} \leftarrow \beta_{SA} P_{SA}, \quad (1)$$