Computers & Industrial Engineering 110 (2017) 216–228

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Biased randomization of heuristics using skewed probability distributions: A survey and some applications

Alex Grasas [a], Angel A. Juan [b,*], Javier Faulin [c], Jesica de Armas [b,d], Helena Ramalhinho [d]

[a] Institute for Transformational Leadership, Spain
[b] Open University of Catalonia, IN3 – Dept. of Computer Science, Spain
[c] Public University of Navarre, Dept. of Statistics and OR, Spain
[d] Univ. Pompeu Fabra, Dept. of Economics and Business, Spain

ARTICLE INFO

Article history:
Received 15 June 2016
Received in revised form 24 January 2017
Accepted 10 June 2017
Available online 13 June 2017

Keywords:
Heuristics
Biased randomization
Real-time decision making
Combinatorial optimization
Logistics
Transportation
Production

ABSTRACT

Randomized heuristics are widely used to solve large scale combinatorial optimization problems. Among the plethora of randomized heuristics, this paper reviews those that contain biased-randomized procedures (BRPs). A BRP is a procedure to select the next constructive 'movement' from a list of candidates in which their elements have *different* probabilities based on some criteria (e.g., ranking, priority rule, heuristic value, etc.). The main idea behind biased randomization is the introduction of a slight modification in the greedy constructive behavior that provides a certain degree of randomness while maintaining the logic behind the heuristic. BRPs can be categorized into two main groups according to how choice probabilities are computed: (i) BRPs using an empirical bias function; and (ii) BRPs using a skewed theoretical probability distribution. This paper analyzes the second group and illustrates, throughout a series of numerical experiments, how these BRPs can benefit from parallel computing in order to significantly outperform heuristics and even simple metaheuristic approaches, thus providing reasonably good solutions in 'real time' to different problems in the areas of transportation, logistics, and scheduling.

1. Introduction

A number of complex decision-making processes in real-life transportation, logistics, and production systems can be modeled as combinatorial optimization problems (Faulin, Juan, Grasman, & Fry, 2012). Among many others, some typical examples include: vehicle routing problems (VRP) (Toth & Vigo, 2014), arc routing problems (Corberán & Laporte, 2014), facility location problems (Chan, 2011), or scheduling problems (Pinedo, 2012). All these problems are NP-hard in nature, meaning that the space of potential solutions grows very fast (exponential explosion) as the instance size increases. Therefore, using exact methods is not always the most efficient strategy, especially when the size of the problem instance is large and reasonably good decisions are needed in negligible computing times. Under these circumstances, heuristic-based approaches constitute an excellent alternative to exact methods (Talbi, 2009). Accordingly, a large number of heuristic and metaheuristic algorithms have been developed during the last decades to solve large scale combinatorial optimization problems and, eventually, support intelligent decision-making processes in a myriad of fields, including transportation, logistics, production, finance, telecommunication, Internet computing, health care, etc.

A constructive heuristic is a computational method that employs an iterative process to generate a feasible solution of reasonable quality. At each iteration of the solution-building process, the next 'movement' is selected from a list of potential candidates that has been sorted according to some criteria. *Pure greedy heuristics* always select the next 'most promising' movement. As a result, these heuristics are expected to generate a reasonably good solution once the entire list is traversed. Notice, however, that this is a somewhat myopic behavior, since the heuristic selects the next movement without considering how the current selection will affect subsequent decisions as the list is processed downwards. Even worse, this property results in a deterministic procedure, i.e., the same solution is obtained every time the algorithm is run. Examples of such methods are the nearest neighbor for traveling salesman problems (Lawler, Lenstra, Rinnooy Kan, & Shmoys, 1985), the shortest processing time dispatching rule for scheduling problems (Pinedo & Chao, 1999), or the savings algorithm for VRPs (Clarke & Wright, 1964). Although these methods are easy to

* Corresponding author.
E-mail addresses: agrasas@iftl.eu (A. Grasas), ajuanp@uoc.edu (A.A. Juan), javier.faulin@unavarra.es (J. Faulin), jesica.dearmas@upf.edu (J. de Armas), helena.ramalhinho@upf.edu (H. Ramalhinho).

http://dx.doi.org/10.1016/j.cie.2017.06.019
0360-8352/

implement and can be run almost instantaneously, the real-time solutions they provide are usually far from being optimal. To improve the quality of these heuristic solutions –and as far as more time is available–, different types of *local search methods* can be used to explore the solution neighborhood (Aarts & Lenstra, 1997). Typically, the neighbor selection is based on a certain logic that tries to take advantage of the specific characteristics of the optimization problem being considered. This usually leads to local optimal solutions. As in the construction phase, if the neighbor chosen is always the next 'most promising' movement according to some criteria, the resulting searching process will be deterministic too.

Randomization techniques are frequently used to escape from this local optimality trap and improve the overall quality of the solution. These techniques can be incorporated either in the construction phase and/or the local search. Randomization allows exploring alternative solutions by selecting an element other than the 'most promising option on the short run'. This leads to different outputs each time the entire procedure is executed. Since running a heuristic might take only a few seconds –or even less in a modern computer if the heuristic is correctly implemented and the instance size is not extremely large–, one can execute it several times, either in sequential mode or in parallel mode by using different threads, and then select the best of the stochastic outputs. Countless metaheuristic algorithms include uniform randomization in their procedures. However, a uniform randomization of the list of candidate elements destroys the logic behind the heuristic greedy behavior. In order to maintain this logic, the randomization can be biased (i.e., oriented) so that higher probabilities are given to the most promising candidates. Thus, the main idea behind biased randomization is the introduction of a slight modification in the greedy constructive behavior that provides a certain degree of randomness while maintaining the main logic behind the heuristic. In a seminal paper on the Monte Carlo method, King (1953) already emphasized the enormous improvement of biasing probabilities on sampling efficiency. Different methods to bias the randomization have been used in multiple contexts thereafter (Fig. 1). Among them, this paper pays special attention to the ones that use skewed (non-symmetric) theoretical probability distributions in order to introduce an appropriate bias in the process of selecting elements from the list during the constructive and/or local search stages. Some skewed theoretical distributions, such as the geometric or the decreasing triangular ones, offer at least two advantages over using empirical distributions: (i) they contain at most one simple parameter, which can be easily set; and (ii) they can be sampled using well-known analytical expressions, which from a computational perspective is typically faster than other sampling techniques involving the use of loops.

In particular, the main contributions of this paper are: (i) to provide a review of the most relevant *biased randomized procedures* (BRPs) used in the literature to solve combinatorial optimization problems; (ii) to provide a general framework for BRPs that use a skewed theoretical probability distribution to bias the selection of the next movement during the constructive and/or local search processes; and (iii) to illustrate, throughout a series of numerical experiments, how these BRPs can significantly outperform heuristics, and even simple metaheuristic approaches, thus providing reasonably good solutions in 'real time' (e.g., one or two seconds) to different transportation, logistics, and scheduling problems.

The remainder of this paper is structured as follows: Section 2 introduces the concept of randomized algorithms; Section 3 presents different BRPs that use empirical bias functions; Section 4 provides a general framework for BRPs with a skewed theoretical probability distribution, and discusses the advantages of this approach over the one based on empirical bias functions; Section 5 analyzes different applications of BRPs to the fields of logistics, transportation, and scheduling; Section 6 describes a series of computational experiments that contribute to illustrate and quantify the potential of BRPs; finally, Section 7 summarizes the main contributions of the paper.

2. Randomized algorithms

There is an enormous body of literature that study probabilistic or randomized algorithms and a review of that is far beyond the scope of this paper. The reader is referred to Collet and Rennard (2006) for a review, and to Clerc (2015) for a vast discussion about the stochastic aspects of optimization. The focus of this paper is in the subset of randomized algorithms that include some type of bias in any of their random processes. A randomized algorithm uses random bits to make random choices during its execution. Unlike deterministic algorithms, different solutions are obtained every time the procedure is executed. The most successful approaches to solve large combinatorial problems take advantage of this feature to perform several iterations and collect the best overall output. These approaches are commonly known as *multi-start methods* (Martí, Resende, & Ribeiro, 2013). In general terms, they all contain two differentiated phases: a construction process and a local or neighborhood search. The former diversifies the search for solutions while the latter intensifies this search. These two phases

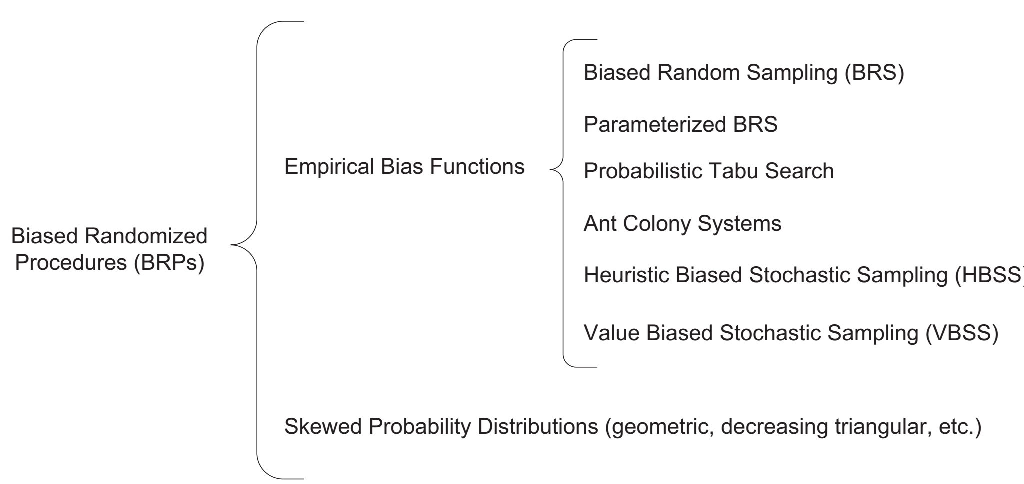

Fig. 1. A classification of biased randomized procedures.