Computers and Operations Research 83 (2017) 106–119

Contents lists available at ScienceDirect

Computers and Operations Research

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

MIP neighborhood synthesis through semantic feature extraction and automatic algorithm configuration

Tommaso Adamo, Gianpaolo Ghiani*, Antonio Grieco, Emanuela Guerriero, Emanuele Manni

Dipartimento di Ingegneria dell'Innovazione, Università del Salento, Via per Monteroni, Lecce, 73100, Italy

A R T I C L E I N F O

Article history:
Received 29 October 2015
Revised 30 January 2017
Accepted 31 January 2017
Available online 9 February 2017

Keywords:
Mixed Integer Programming
Neighborhood search
Semantic features

A B S T R A C T

The definition of a “good” neighborhood structure on the solution space is a key step when designing several types of heuristics for *Mixed Integer Programming* (MIP). Typically, in order to achieve efficiency in the search, the neighborhood structures need to be tailored *not only* to the specific problem *but also* to the peculiar distribution of the instances to be solved (*reference instance population*). Nowadays, this is done by human experts through a time-consuming process comprising: (a) problem analysis, (b) literature scouting and (c) experimentation. In this paper, we illustrate an *Automatic Neighborhood Design* algorithm that mimics steps (a) and (c). Firstly, the procedure extracts some semantic features from a MIP compact model. Secondly, these features are used to derive automatically some neighborhood design mechanisms. Finally, the “proper mix” of such mechanisms is sought through an *automatic configuration phase* performed on a training set representative of the reference instance population. When assessed on four well-known combinatorial optimization problems, our automatically-generated neighborhoods outperform state-of-the-art model-based neighborhoods with respect to both *scalability* and *solution quality*.

© 2017 Published by Elsevier Ltd.

1. Introduction

The definition of a “good” neighborhood structure on the solution space is a key step when designing several families of metaheuristics for combinatorial optimization problems. It is well known that neighborhood structures need to be tailored to the specific problem in order to make the search efficient. Indeed, the neighborhood design should also consider the peculiar distribution of the instances to be solved (*reference instance population*): for example, when designing a heuristic for the *Vehicle Routing Problem with Time Windows* (Toth and Vigo, 2014), the choice of a neighborhood structure depends on whether the instances are characterized by large or tight time windows. Other aspects to be considered are: the algorithm (either exact or heuristic) used to explore the neighborhoods; the time limit imposed on the exploration of a neighborhood; the computer hardware on which the search algorithm runs.

Nowadays, the design of a neighborhood structure is done by human experts through a time-consuming and iterative process comprising three steps: (a) problem analysis; (b) literature scouting and (c) experimentation. During phase (a), the experts examine and characterize the problem structure as well as the particular distribution of the instances to be solved; at this stage, properties are derived and meaningful features are identified. In step (b), the literature is searched in the hope of finding successful algorithms developed previously by other researchers for the same or similar problems. At stage (c), a number of tentative neighborhood structures are tested on a *training set* extracted from the reference instance population. It is often the case that these tentative designs are parameterized and some sort of experimental design approach (Montgomery, 2012) is used to select the best parameter values. As a rule, steps (a), (b) and (c) are not performed sequentially. For instance, a preliminary problem analysis may help selecting relevant work in the scientific literature. In turn, the findings of this study may suggest the derivation of further properties of the instances to be solved, in order to choose the best algorithmic approach among those reported in the literature.

We aim to develop mechanisms that may derive automatically, without any human intervention, suitable neighborhoods on the basis of a *Mixed Integer Programming* (MIP) model of the problem. In particular, we illustrate in this paper a procedure that mimics steps (a) and (c). Our *Automatic Neighborhood Design* (AND) algorithm takes as input a MIP model and a training set (representative of the reference instance population), and provides a *neighborhood structure* $N()$ as output (Fig. 1). A neighborhood structure $N()$ is itself a procedure that associates a subset of

* Corresponding author.
E-mail address: gianpaolo.ghiani@unisalento.it (G. Ghiani).

http://dx.doi.org/10.1016/j.cor.2017.01.021
0305-0548/© 2017 Published by Elsevier Ltd.

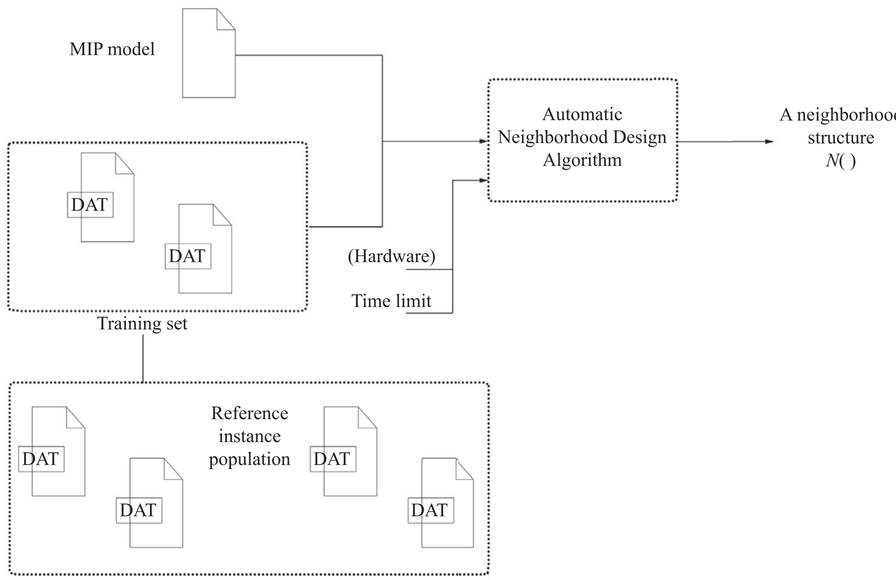

Fig. 1. Schematic input-output representation of an *Automatic Neighborhood Design* algorithm.

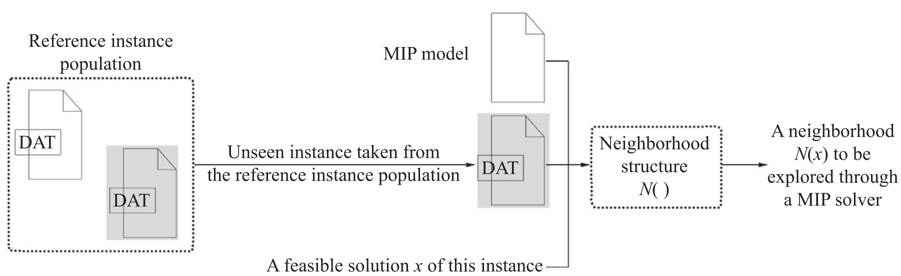

Fig. 2. Utilization of an *automatically designed neighborhood structure*.

feasible solutions $N(x)$, called a *neighborhood*, to any solution x of the reference instance population (Fig. 2).

The ultimate goal of the research is to automatically generate neighborhood structures that, when embedded into a metaheuristic, may provide *human competitive* results, i.e. solutions that are comparable to, or better than, the solutions generated - with the same amount of computational resources - by state-of-the art algorithms (designed by human experts). See Koza (2010) for an in-depth discussion of this concept. Unfortunately, we are not at that stage yet (as indicated in Section 6), although the AND procedure presented in this paper outperforms previous domain-independent neighborhoods with respect to both *scalability* and *solution quality*.

The remainder of the paper is organized as follows. In Section 2, we review the literature relevant to our work while in Section 3 we describe the basic idea underlying our approach. In Section 4, we present a procedure that automatically extracts some semantic features from a MIP model and a given current feasible solution. In Section 5 we illustrate an algorithm that - based on the previously extracted features - identifies some neighborhood design mechanisms. We also show how to make use of an *Automated Algorithm Configuration* procedure to automatically choose the "best" mix of such mechanisms over the reference instance population. In Section 6 we present computational results on four well-known combinatorial optimization problems. These results highlight the improvements provided by our automatically-designed neighborhoods with respect to state-of-the-art domain-independent neighborhoods. Conclusions follow in Section 7.

2. Literature review

We now review the most relevant literature related to our work. First of all, it is worth mentioning that there are several algorithms for general MIPs (such as the Feasibility Pump with all its variants and improvements (Fischetti et al., 2005), (Achterberg and Berthold, 2007), (Bertacco et al., 2007; Fischetti and Salvagnin, 2009) or the Kernel Search (Angelelli et al., 2010), (Angelelli et al., 2012), (Guastaroba and Speranza, 2012)) that allow to obtain a first feasible solution from a MIP model. There also exist generic heuristic black-box solvers (like LocalSolver Benoist et al. (2011)) in which a problem is modeled using a specific formalism and its resolution is deferred to a solver based on some local search techniques. Here, we take a different perspective. Indeed, we aim to improve a given feasible solution by means of a machine-generated neighborhood structure. Along this line of research, Fischetti and Lodi (2003) proposed the *Local Branching* algorithm which utilizes *spherical neighborhoods* defined by appropriate non valid inequalities. For purely binary MIPs, a neighborhood is made up of all solutions with a *Hamming distance* from the current solution that does not exceed a given value. The neighborhoods are then explored by a generic black-box MIP solver. The Local Branching paradigm has subsequently been refined and extended. In particular, Hansen et al. (2006) combined Local Branching with *Variable Neighborhood Search* (VNS), whereas Lazić et al. (2010) proposed a hybrid heuristic for solving 0–1 MIPs based on the principle of *Variable Neighborhood Decomposition Search*.

Download English Version:

https://daneshyari.com/en/article/4959013

Download Persian Version:

https://daneshyari.com/article/4959013

Daneshyari.com