A Branch-and-Bound-Based Crossover Operator for the Traveling Salesman Problem

Thomas Weise, Institute of Applied Optimization, Hefei, China
Yan Jiang, University of Science and Technology of China, Shanghai, China
Qi Qi, University of Science and Technology of China, Hefei, China
Weichen Liu, University of Science and Technology of China, Hefei, China

ABSTRACT

In this article, the new crossover operator BBX for Evolutionary Algorithms (EAs) for traveling salesman problems (TSPs) is introduced. It uses branch-and-bound to find the optimal combination of the (directed) edges present in the parent solutions. The offspring solutions created are at least as good as their parents and are only composed of parental building blocks. The operator is closer to the ideal concept of crossover in EAs than existing operators. This article provides the most extensive study on crossover operators on the TSP, comparing BBX to ten other operators on the 110 instances of the TSPLib benchmark set in EAs with four different population sizes. BBX, with its better ability to reuse and combine building blocks, surprisingly does not generally outperform the other operators. However, it performs well in certain scenarios. Besides presenting a novel approach to crossover on the TSP, the study significantly extends and refines the body of knowledge on the field with new conclusions and comparison results.

KEYWORDS

BBX, EA, Evolutionary Algorithm, Metaheuristic, Optimization, Recombination, TSP, TSPLib

1. INTRODUCTION

Evolutionary Algorithms (EAs; Goldberg, 1989; Weise, 2009) are maybe the most successful family of metaheuristics and a prime example of the application of natural intelligence to optimization, i.e., to the solution of hard problems. When first introduced in the 70s of the last century by Holland (1975), they provided two major innovations in the field of optimization: populations for guarding against premature convergence and binary search operators (also called crossover or recombination). The general idea behind the binary search operator is that its input is two (different) solutions which have been selected, i.e., which are good. Combining these solutions would hopefully lead to a new solution which unites the positive aspects of both parents to attain even better quality. We can divide crossover operators into three families as follows.

DOI: 10.4018/IJCINI.2019070101

Copyright © 2019, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
A first family of operators simply copies parts of the two “parents” into the offspring, without using any direct knowledge of the problem instance or objective function. If the parent solutions are reasonably good, such an unbiased sampling from parents is more likely to construct worse than better offspring solutions. Thus, the selection taking place in the EA will be the driving force of the optimization process.

A second family of operators incorporates problem instance-specific knowledge. For example, a heuristic can be used to efficiently combine elements of the parent solutions. A new solution may be constructed by iteratively choosing and attaching the seemingly most suitable building block from each of the two parents. Such a heuristic crossover idea adds another driving force to the optimization procedure. It biases the search direction and makes the EA greedier.

Recently, a third family of operators has emerged: operators that attempt to find the best possible combination of parental building blocks by applying exact approaches. We introduce a new member of this family, BBX, which uses a fast and deterministic Branch-and-Bound (BB) algorithm. A globally optimal solution is one which cannot be outperformed by any other solution. Exact solvers like BB can find such solutions, but, due to the $NP$-hardness of the TSP, take too long to do so. EAs and metaheuristics in general trade the guarantee of optimal solution quality in for a shorter runtime. They return approximate solutions which may be worse than the best tours. Our BBX operator constructs solutions which represent the optimal combination of the building blocks of their parents. These are usually not globally optimal, but enable the EA to jump from two good solutions to another one.

Whether such an operator can be efficient depends on several issues: First, the improved solution quality compared to purely randomized operators needs to be worth the potentially higher runtime requirement. Second, the operator needs to integrate well with the rest of the EA. Limiting the offspring components strictly to elements from the parent solutions will necessarily decrease the diversity (Weise, Chiong, & Tang, 2012; Weise, 2009) and may lead to premature convergence.

The heuristic operators from the second family often greedily combine building blocks from parent solutions and, if reaching an impasse, generate new genetic material to complete the offspring. The exact operators in the third family are the greediest possible operators but will never create new building blocks. In summary, the idea of a perfect binary search operator, which ideally fulfills the concept of crossover and returns the perfect combination of parental building blocks is very tempting. However, it must be analyzed whether it actually improves the solution quality and runtime behavior of the EA.

The key contributions of this work to the research on natural intelligence and EAs for the TSP are:

1. A new BBX operator is defined, which ideally combines (directed) edges of parental solutions using Branch-and-Bound;
2. The largest study on eleven crossover operators for the TSP to date, is conducted, including problems ranging from 14 to 85'900 cities;
3. We draw conclusions with general and important implications regarding the complexity of EA behavior and recommendations to experimental design.

In the remainder of this paper, we first provide some background information in Section 2. Here we introduce the Traveling Salesman Problem (TSP) (Applegate, Bixby, Chvátal, & Cook, 2007; Lawler, Lenstra, Rinnooy Kan, & Shmoys, 1985; Gutin & Punnen, 2002), which we use as testbed for our investigation. We also discuss related work on crossover operators, both from the perspective of our above-defined operator families and the state-of-the-art operators for the TSP and emphasize the importance of runtime behavior analysis over end result comparison in experimentation with optimization methods in this section. We then introduce our new crossover operator from the third operator family in Section 3 by first describing the underlying BB algorithm in detail and then its adaptation for crossover. We analyze the performance of the new operator in a large-scale experimental
Related Content

Visualization by Coding: Drawing Simple Shapes and Forms in Various Programming Languages
[www.igi-global.com/chapter/visualization-by-coding/127482?camid=4v1a](www.igi-global.com/chapter/visualization-by-coding/127482?camid=4v1a)

A Novel Approach for Designing Dynamical S-Boxes Using Hyperchaotic System
[www.igi-global.com/article/novel-approach-designing-dynamical-boxes/67796?camid=4v1a](www.igi-global.com/article/novel-approach-designing-dynamical-boxes/67796?camid=4v1a)
Design of Cognitive Machines
www.igi-global.com/chapter/design-cognitive-machines/27874?camid=4v1a

Computational Creativity: Improv Agents and Conceptual Blending
Rania A. Hodhod and Brian S. Magerko (2014). International Journal of Cognitive Informatics and Natural Intelligence (pp. 1-14).
www.igi-global.com/article/computational-creativity/126264?camid=4v1a