Round-Table Group Optimization for Sequencing Problems

Xiao-Feng Xie, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

ABSTRACT

In this paper, a round-table group optimization (RTGO) algorithm is presented. RTGO is a simple meta-heuristic framework using the insights of research on group creativity. In a cooperative group, the agents work in iterative sessions to search innovative ideas in a common problem landscape. Each agent has one base idea stored in its individual memory, and one social idea fed by a round-table group support mechanism in each session. The idea combination and improvement processes are respectively realized by using a recombination search (XS) strategy and a local search (LS) strategy, to build on the base and social ideas. RTGO is then implemented for solving two difficult sequencing problems, i.e., the flowshop scheduling problem and the quadratic assignment problem. The domain-specific LS strategies are adopted from existing algorithms, whereas a general XS class, called socially biased combination (SBX), is realized in a modular form. The performance of RTGO is then evaluated on commonly-used benchmark datasets. Good performance on different problems can be achieved by RTGO using appropriate SBX operators. Furthermore, RTGO is able to outperform some existing methods, including methods using the same LS strategies.

Keywords: Global Optimization, Group Creativity, Idea Combination Process, Meta-Heuristic Framework, Recombination Search, Sequencing Problems, Social-Biased Learning

1. INTRODUCTION

Group creativity techniques, e.g., brainstorming (Osborn, 1953), have been widely studied in social science (Nemeth, 1986; Paulus, 2000). A cooperative group contains multiple individuals who have some interactions on ideas of each other’s (Paulus, 2000), and the group tries to find innovative solutions (or high-quality ideas) for a specific task by generating new (especially innovative) ideas spontaneously contributed by its members in iterative idea-generating sessions.

Group creativity has been studied in both group and individual levels. At the group level, a support mechanism is used to assist individuals by providing useful external stimulus in their idea-generation process. The essential function of these mechanisms is to serve as group memory (Betts & Hinsz, 2010; Satzinger, Garfield, & Nagasundaram, 1999) for the diffusion of innovative patterns (Rogers, 2003) as well as providing diverse ideas (Paulus, 2000), based on a developing repository of nonprivate knowledge shared by members. For individuals, group memory might be heterogeneously shaped in network structures (Lovejoy & Sinha, 2010; Rulke & Galaskiewicz, 2000).

For each individual, the ideation process involves idea selection (Rietzschel, Nijstad, & Stroebe, 2010; Putman & Paulus, 2009) and idea
generation (Nijstad & Stroebe, 2006; Paulus, 2000; Kohn, Paulus, & Choi, 2011), based on its individual memory (Glenberg, 1997; Newell & Simon, 1972). A pre-selection mechanism (Putman & Paulus, 2009) might be used to pick out relevant ideas from both the individual and group memory. Then an idea-generation mechanism is used to build new idea(s) on selected ideas. Afterward, a post-selection mechanism (Rietzschel et al., 2010) is applied to update the individual memory. Each individual also holds a sharing mechanism to contribute nonprivate knowledge (Liu & Tsui, 2006) for the group.

The idea-generation process is a critical part of the creative process. Since the research of Osborn (1953), combine-and-improve ideas has been a general brainstorming rule to form a single better idea by building on existing ideas. The total process of building on existing ideas might be divided into two parts, the idea combination process (Kohn et al., 2011; Yu & Nickerson, 2011) and the idea improvement process, roughly corresponding to the divergent and convergent processes, where the former is of paramount importance although the latter is also nontrivial (Cropley, 2006).

Human problem solving can be seen as searching in a structural space of states (ideas) (Newell & Simon, 1972). Thus, there is a natural similarity between creative idea-generating tasks and hard optimization problems (Lovejoy & Sinha, 2010), both of them require efficiently achieving high-quality solutions.

Many optimization problems can be formulated as sequencing (or permutation) problems (Koivisto & Parviainen, 2010; Starkweather et al., 1991). All sequencing problems with \( n \) nodes share the same problem space, in which each potential solution (or state) \( \bar{x} \) is a permutation \( \{x_1, \ldots, x_n\} \) of the integer values from 1 through \( n \), only their objective functions \( f(\bar{x}) \) possess different structural properties.

In this paper, we consider two typical sequencing problem examples, i.e., the quadratic assignment problem (QAP) (Burkard, Karisch, & Rendl, 1997; Taillard, 1990) and the flowshop scheduling problem (FSP) (Beasley, 1990; Reeves, 1995; Taillard, 1993), among some other examples (Xie, Smith, Lu, & Barlow, 2011). Both QAP and FSP are NP-hard in the strong sense. FSP is a well-known problem in intelligent manufacturing systems. QAP arises in many practical applications, e.g., design of grey patterns and website structure improvement (Saremi, Abedin, & Kermani, 2008). QAP also serves as a generalization of some other important optimization problems (Merz & Freisleben, 2000).

Exact algorithms, e.g., branch-and-bound (Carlier & Rebai, 1996), can only be tractable for solving small-scale instances, whereas fast constructive heuristics (Reeves, 1985; Nawaz, Enscore, & Ham, 1983) often obtain results that are far away from optimal. Thus, low-level search components, especially local search (LS) and recombination search (XS), as well as upper-level meta-heuristic frameworks, have been integrated together for finding near optimal solutions within practical computational costs.

Each LS strategy improves an incumbent solution by intensively moves based on some neighborhood search operations (Gambardella, Taillard, & Dorigo, 1999), e.g., 2-opt. A LS strategy is defined as stable if it only allows non-worse moves. Any stable LS, such as hill-climbing and the fast LS (Gambardella et al., 1999), cannot escape from the local minimum it first encounters. Some advanced LS strategies, such as simulated annealing (Nearchou, 2004b), tabu search (Nowicki & Smutnicki, 1996; Taillard, 1991), iterated local search (Stutzle, 1998; Ruiz & Stutzle, 2007), etc., incorporate some unstable moves so as to explore in a rugged problem landscape.

Each XS strategy generates a new solution by preserving positive clues in two parent solutions, which has an implicit advantage of adaptive leaping by utilizing the difference between two parents (Merz & Freisleben, 2000). Typical examples of XS operators for sequencing problems include order crossovers (Murata, Ishibuchi, & Tanaka, 1996; Starkweather et al., 1991), LCS crossover (Iyer & Saxena, 2006).
www.igi-global.com/chapter/parameter-optimization-of-photovoltaic-solar-cell-and-panel-using-genetic-algorithms-strategy/147530?camid=4v1a

Hybrid Optimization Techniques for Industrial Production Planning: A Review
www.igi-global.com/chapter/hybrid-optimization-techniques-for-industrial-production-planning/82688?camid=4v1a