Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Discrete Mathematics 36 (2010) 939–946
Electronic Notes in DISCRETE MATHEMATICS
www.elsevier.com/locate/endm

Multiple crossover genetic algorithm for the multiobjective traveling salesman problem

Semya Elaoud [1]

LOgistique Gestion Industrielle et de la Qualité (LOGIQ)
Institut Supérieur de Gestion Industrielle
Sfax, Tunisie

Jacques Teghem [2]

Service de Mathématique et de Recherche Opérationnelle (MathRO)
Faculté Polytechnique de Mons
Mons, Belgique

Taïcir Loukil [3]

LOgistique Gestion Industrielle et de la Qualité(LOGIQ)
Institut Supérieur de Gestion Industrielle
Sfax, Tunisie

Abstract

Many crossover operators have been proposed and adapted to different combinatorial optimization problems. In particular, many permutation based crossovers are well designed for the traveling salesman problem (TSP) which is among the most-studied combinatorial optimization problems. However, there is no evidence that one crossover operator is superior to another operator. This is specially true for multiobjective optimization. The performance of any genetic algorithm generally varies according to the crossover and mutation operators used. We propose to include mutiple crossover and mutation operators with a dynamic selection scheme

doi:10.1016/j.endm.2010.05.119

into a multiobjective genetic algorithm in order to choose the best crossover operator to be used at any given time. The objective is to find a good approximation of the Pareto set. Experimental results on different benchmark data show synergy effects among different used crossovers and prove the efficiency of the proposed approach.

Keywords: Genetic algorithm, multiobjective optimization, traveling salesman problem

1 Introduction

Genetic algorithms are very powerful methods for multiobjective optimization problems and have been successfully applied to various problems [4]. The interesting behavior arises from genetic algorithms because of the ability of solutions to learn from each other. Their performance, then, highly depends on encoding scheme and the choice of genetic operators, especially crossover and mutation operators.

Over decades, researchers have suggested a large variety of crossover and mutation operators. In particular, several operators have been developed and adapted to the permutation representation that may be used for a wide variety of combinatorial optimization problems. A typical example and one of the most studied problems in this field is the traveling salesman problem (TSP).

The single objective TSP is classified as NP-Hard problems. Multiobjective version is even harder since it has the difficulty of the TSP itself and the difficulty of multiple objectives [3]. It requires obtaining a set of diverse and nondominated solutions forming the Pareto front, rather a single solution.

Different solution encodings may be used to solve the TSP when using genetic algorithms such as binary representation and matrix representation. The most appropriate one is the permutation representation in which, the n cities that should be visited are put in order according to a list of n elements, so that if the city i is the j^{th} element of the list, city i is the j^{th} city to be visited [7].

These operators differ in the way to explore the solution space and generate new solutions. They have different ways to perform the exchange of information between individuals during the search. Typically, one crossover and one mutation operator are used in any single genetic algorithm. However

[1] Email: samyaelaoud@yahoo.fr
[2] Email: Jacques.Teghem@umons.ac.be
[3] Email: Taicir.Loukil@fsegs.rnu.tn

there is no obvious way either to set the best crossover and mutation rates nor to select best operators. The choice is generally determined by multiple runs with different combinations and extensive experiments. But, does the combination of two or more operators enhance any interaction effects? Do combined operators cooperate to provide better performance than each one used separately? Is an operator better at the beginning of the algorithm, when few non dominated solutions are available, and another operator better at the end of the algorithm, when it is more difficult to generate new dominated solutions?

In this paper, we propose to empirically study the synergy of multiple crossover and mutation operators for solving the multiobjective traveling salesman problem. Motivated by successful applications of the Pareto Fitness Genetic Algorithm (PFGA) on both real world and theoretical problems [1][2], we included in PFGA multiples crossover and mutations operators with a dynamic selection scheme through which the algorithm decides which is the best operator -or combination crossover/ mutation operators- to be used at any given time.

2 Method description

We propose to include multiple crossover and mutation operators with a dynamic selection scheme into the Pareto Fitness Genetic Algorithm (PFGA) in order to choose the best operators to be used at any given time.

2.1 The PFGA

The Pareto Fitness Genetic Algorithm (PFGA) is our recent contribution in the field of multi-objective evolutionary algorithms [1]. The algorithm makes use of a double ranking strategy where to an individual is, in a first stage, assigned a dummy rank value representing the number of individuals that dominate it. An individual's rank is then defined as the summation of its own dummy rank value and those of its dominators. The algorithm include a promising sharing procedure called Population-size Adaptive Density Estimation strategy (PADE): the search space is dynamically divided into a given number of cells and the density value of a given individual is then the number of solutions located in the same cell. A Pareto fitness function based on the rank of the individual and its density value is used to evaluate solutions (for more details on (PFGA) see [1]).

The PFGA has been tested on various types of continuous multi-objective benchmark problems and compared with state-of-the-art algorithms. It has been shown to be an efficient approach providing a good compromise between the quality of approximated solutions and the required computational time [1]. The PFGA has also been combined with a linear programming solver in order to solve both single and mutiobjective version of a particular Mixed Integer Programming problem [2].

2.2 Probability adaptation

Several crossover and mutation operators are considered.
We denote by:

m : the number of potential crossover operators.
n : the number of potential mutation operators.
$\mathscr{P}$: the set of all operators combinations or pairs.

- ∘ any crossover + any mutation operators
- ∘ any crossover operator + no mutation
- ∘ no crossover + any mutation operator

such that:

$$|\mathscr{P}| = [(m+1) \times (n+1)] - 1 \tag{1}$$

The idea is to choose randomly, at each iteration, a pair s of $\mathscr{P}$ to generate the next population. To entail an improvement on the global performance of the whole population, the probability of each pair to be chosen will be modified at each iteration depending on the interest of the results produced by this iteration.
The initial probability of any pair to be chosen is set to :

$$p_i = \frac{1}{|\mathscr{P}|} \tag{2}$$

A pair $s \in \mathscr{P}$ is randomly chosen to generate offspring. The probability p_s will then be decreased or increased depending on the resulted solution quality, and the other probabilities p_i, $i \neq s$, adapted accordingly.

A measure of the interest $\Delta_n \geq 0$ of the selected pair s is then computed for each iteration n. This crucial measure will be defined below. The probabilities are modified before the next iteration such that:

- if $\Delta_n > 0 : p_s \nearrow$ and $p_i \searrow,\ i \neq s$
  - ▷ $p_s \leftarrow p_s + (1 - p_s)\Delta_n$
  - ▷ $p_i \leftarrow p_i - p_i\Delta_n,\ i \neq s$
- if $\Delta_n = 0 : p_s \searrow$ and $p_i \nearrow,\ i \neq s$ (see below)

To define Δ_n, we denote (avoiding index n by facility) :
N: Cardinality of the population
E: Elite set of potential efficient solutions
S_E: Set of new solutions entering in E.
$S_E = S_e \cup S_d$
S_e : set of new solutions entering in E without domination of any solution of E.
S_d : set of new solutions entering in E with domination of at least one solution of E.
S_r: Set of solutions rejected from E (in case of $|S_d| > 0$)
The measure of interest Δ_n of the selected pair is then defined as follows:

$$\Delta_n = \beta_n \times Min\left\{1,\ \frac{|S_E| + |S_r|}{N}\right\} \tag{3}$$

with $\beta_n < 1$, maximal value of Δ_n at iteration n, defined by:

$\beta_n = 1 - \alpha^n$ where α, a parameter of the method, is very close to 1.

We present in the following a deeper analysis of possible values of the measure of interest Δ_n : $0 \leq \Delta_n \leq \beta_n$

- $\Delta_n = \beta_n$ if $|S_E| + |S_r| \geq N$
- $\Delta_n = \beta_n \times \frac{x}{N}$ with $x < N$, for instance if:

$$\left.\begin{array}{l} \triangleright\ |S_e| = x, \quad |S_d| = 0 \quad (= |S_r|) \\ \triangleright\ |S_e| = 0, \quad |S_d| = 1 \quad |S_r| = x - 1 \end{array}\right\} \text{ i.e. x changes in E.}$$

- $\Delta_n = 0$ iff $|S_E| = 0$ $(= |S_r|)$
  In this case, we propose to adapt the probabilities associated with each pair in the following way:
  $\triangleright\ p_s \leftarrow p_s - p_s \times (1 - \beta_n) \times \frac{1}{N}$
  $\triangleright\ p_i \leftarrow p_i + p_s \times (1 - \beta_n) \times \frac{1}{N \times (|\mathscr{P}|-1)},\ i \neq s$

3 Experimental analysis

In order to study the synergy of multiple operators, we applied the algorithm to a well known set of instance that was tackled by several authors [10][8] [5]. Seven of the best known crossover operators are considered and experimented: the Partially mapped crossover (PMX), the Order crossover (OX), the Cycle crossover (CX), the Order based crossover (OBX), the Position based crossover (PBX), the Heuristic crossover (HX) and Edge recombination

	PMX	OX	CX	OBX	POS	HX	ER	NX
DM	0.0144	0.0138	0.0201	0.0148	0.0159	0.0135	0.0197	0.0148
IVM	0.0197	0.0148	0.0181	0.0228	0.0206	0.0242	0.0144	0.0155
EM	0.0144	0.0155	0.0232	0.0156	0.0456	0.0200	0.0195	0.0159
ISM	0.0195	0.0159	0.0155	0.0181	0.0150	0.0148	0.0151	0.0164
SIM	0.0151	0.0164	0.0154	0.0161	0.0157	0.0146	0.0165	0.0163
NM	0.0165	0.0163	**0.0306**	0.0160	**0.0443**	**0.0417**	**0.0543**	

Table 1
50×2 Final probabilities

	PMX	OX	CX	OBX	POS	HX	ER	NX
DM	0.0232	0.0217	0.0232	0.0217	0.0033	0.0232	0.0178	0.0195
IVM	0.0232	0.0232	0.022	0.0232	0.0176	0.0231	0.0232	0.023
EM	0.017	0.0134	0.0093	0.0109	0.0232	0.0232	0.0228	0.0034
ISM	0.0148	0.0116	0.0018	0.0178	0.0232	0.0160	0.0043	0.0176
SIM	0.0023	0.0016	0.023	0.0222	0.0232	0.0136	0.0137	0.022
NM	0.023	0.0119	0.0231	0.0212	0.0232	0.0232	**0.1908**	$\times$

Table 2
100×2 Final probabilities

crossover (ERX); as well as five of the best known mutation operators: the Displacement mutation (DM), the Inversion mutation (IVM), the Exchange mutation (EM), the Insertion mutation (ISM) and the Simple insertion mutation (SIM). For a detailed review of different genetic operators see [7]. NX(NM) represents the possibility not to use any crossover (mutation) operator. At each iteration a Roulette wheel selection is applied to choose one pair among 47 combinations of operators. This pair is used to generate solution of the whole next generation. The probabilities are then updated according to the quality of produced solutions as detailed previously.

Selected instances are classified into three main classes according to the size and the objectives number: $(a \times b)$ denotes a set of 10 instances with a cites and d objectives.

3.1 Evolution of the probabilties

Mean values of final probabilities for three different classes are reported in tables 1- 3. (highest probabilities are indicated in bold)

A first obvious conclusion is that pairs not including mutations operators perform in general better than those including a mutation operator. The reason is probably that enough diversity is provided by the variation of the crossover operator. Therefore we pay more attention in our discussion to the role of crossovers. The ER/NM showed two peaks of probability in both

	PMX	OX	CX	OBX	POS	HX	ER	NX
DM	0.0077	0.0071	0.0077	0.0077	0.0078	0.0077	0.0076	0.0077
IVM	0.0077	0.0077	0.0078	0.0077	0.0077	0.0072	0.0077	0.0077
EM	0.0077	0.0077	0.0076	0.0077	0.0077	0.0077	0.0077	0.0077
ISM	0.0072	0.0077	0.0077	0.0077	0.0076	0.0077	0.0081	0.0077
SIM	0.0077	0.0077	0.0069	0.0077	0.0077	0.0077	0.0077	0.0077
NM	0.0079	0.0077	0.0077	0.0077	0.0077	0.0077	**0.6477**	×

Table 3
100×3 Final probabilities

(100×2) and (100×3) and share highest probabilities with POS/NM, HX/NM and CX/NM in (50×2) instance set. Larranaga et al. [7] argued that crossover operators try to pass on two types of information to the offspring when applied on the TSP: the absolute position of the cities in the tours (such as CX) and the relative order of the cities in the parents tours (such as ER). In particular, the ER transmits (in average) 30% parents egdes when the probelm size is relatively important. Indeed, solving larger instances necessarily requires higher conservation of adjacency relations, especially in the final process stages.

probabilities of the (100×3) seems however to be more regular. This can be explained by the fact that one a crossover is selected the number of accepted solution is very huge. It is easier to generate new elite solution due to the huge number of the Pareto set solution. Hence, any operator always get a $\Delta_n > 0$ which makes his probability stronger. However, due to the the probability adaptation strategy only the most efficient crossover can be distinguished. It is obvious here that the best performance crossover is the one how provide convergence advantage to the Pareto set. Indeed the difference is generated by the number of eliminated elite solutions. The better a solution is the bigger the number of eliminated solutions.

4 Conclusion and Future researches

This paper points out synergic effect resulted from the use of multiple crossover operators. Intensive experimental results showed that an operator may perform efficiently at the beginning of the search and another operator may be more useful at final stages. Further researches can be made in different directions: (i) Deeper computational experiments could be done, in particular comparison with the recent results of the two phase local search methods ([8] and [10]) for the biobjective TPS. (ii) The synergy of different subsets of the operators or other probability adaptation strategies (definition of Δ_n and β_n) could be tested. (iii) Comparison between the use of one and multiple crossovers could also be useful to evaluate different combinations.

References

[1] Elaoud S, Loukil T, Teghem J (2007) A Pareto Fitness Genetic Algorithm: test function study. European Journal Operational Research, 177 (3), 1703-1719.

[2] Elaoud S, Teghem J, Bouaziz B (2007) Genetic Algorithms to solve the cover printing problem. Computers and Operations Research, 34 (11), 3346-3361.

[3] Ehrgott M, (2000) Approximation algorithms for combinatorial multicriteria optimization problems. International Transactions in Operational Research, 7:5-31.

[4] Ehrgott M, and Gandibleux X (2002) Multiobjective Combinatorial Optimization - theory, methodology, and applications. In: Ehrgott M, Gandibleux X (eds.)Multiple Criteria Optimization: State of the Art. Annotated Bibliographic Survey .Kluwer's International Series in Operations Research and Management Science, 52: 369-444. Kluwer Academic Publishers, Boston. 2002. ISBN 1-4020-7128-0.

[5] Jaszkiewicz A (2002) Genetic local search for multi-objective combinatorial optimization. European Journal of Operational Research, 137 (1), 50-71.

[6] Kumar R, Singh P K (2007) Pareto Evolutionary Algorithm Hybridized with Local Search for Biobjective TSP. Studies in Computational Intelligence (SCI) 75 : 361-398.

[7] Larranaga P, Kuijpers C M H, Murga R H, Inza I, Dizdarevic S (1999) Genetic algorithm for the Traveling Salesman problem: A review of representation and operators. Artificial Intelligence Review, 13 (2): 129-170.

[8] Lust T, Teghem J MEMOTS (2008) A memetic algorithm integrating tabu sarch for combinatorial multiobjective optimization. RAIRO, 42, 3-33.

[9] Toscano Pulido G, Coello Coello C A (2003) The Micro Genetic Algorithm 2: Towards Online Adaptation in Evolutionary Multiobjective Optimization. In: Fonseca C M, Fleming P J, Zitzler E, Deb K, Thiele L (eds.) Proceedings of the second Evolutionary Multi-Criterion EMO 2003, 252-266. Faro, Portugal.

[10] Paquete L, Stützle T (2003) A Two-Phase Local Search for biobjective Traveling Salesamn Problem. In: Fonseca C M, Fleming P J, Zitzler E, Deb K, Thiele L. (eds.) Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, 479-493, Faro, Portugal.