European Journal of Operational Research xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms

Shuai Yuan *, Bradley Skinner, Shoudong Huang, Dikai Liu

School of Electrical, Mechanical and Mechatronic Systems, Faculty of Engineering and Information Technology (FEIT), University of Technology, Sydney, PO Box 123, NSW 2007, Australia

ARTICLE INFO

Article history:
Received 17 November 2011
Accepted 19 January 2013
Available online xxxx

Keywords:
Two-part chromosome
Crossover operator
MTSP
Genetic algorithms

ABSTRACT

This paper proposes a new crossover operator called two-part chromosome crossover (TCX) for solving the multiple travelling salesmen problem (MTSP) using a genetic algorithm (GA) for near-optimal solutions. We adopt the two-part chromosome representation technique which has been proven to minimise the size of the problem search space. Nevertheless, the existing crossover method for the two-part chromosome representation has two limitations. Firstly, it has extremely limited diversity in the second part of the chromosome, which greatly restricts the search ability of the GA. Secondly, the existing crossover approach tends to break useful building blocks in the first part of the chromosome, which reduces the GA's effectiveness and solution quality. Therefore, in order to improve the GA search performance with the two-part chromosome representation, we propose TCX to overcome these two limitations and improve solution quality. Moreover, we evaluate and compare the proposed TCX with three different crossover methods for two MTSP objective functions, namely, minimising total travel distance and minimising longest tour. The experimental results show that TCX can improve the solution quality of the GA compared to three existing crossover approaches.

1. Introduction

The multiple travelling salesman problem (MTSP) is a variation of the well-known travelling salesman problem (TSP), which is classed as being NP-hard. The MTSP is more difficult than the TSP, because it aims to resolve a group of Hamiltonian circuits without sub-tours for m ($m > 1$) salesmen to serve a set of n ($n > m$) cities. This leads to the optimum solution of MTSP becoming more computationally infeasible as the problem size increases. Compared to the TSP, the MTSP is more suitable for modelling many practical situations because it is capable of handling more than one salesman. As such, many practical problems have been modelled as a MTSP, such as print press scheduling (Gorenstein, 1970), crew scheduling (Svestka and Huckfeldt, 1973), hot rolling scheduling (Tang et al., 2000), mission planning (Ryan et al., 1998) and vehicle scheduling (Park, 2001). Additionally, there are several variations of the classical MTSP. Based on the number of depots, the MTSP can have a single depot or multiple depots. It can also have open or closed tours, where the difference is whether the salesmen need to return to their depot(s). Furthermore, if the salesmen need to pick up loads, it can be cast as the MTSP pickup and delivery problem (MTSPPDP) (Wang and Regan, 2002). If the tasks have time-related constraints, the MTSP is cast with time windows (MTSPTW) (Kim and Park, 2004). There are other variations of the MTSP in real world applications; however, in all these MTSP applications, job planning and vehicle scheduling are the most commonly researched areas (Bektas, 2006).

Due to the combinatorial complexity of the MTSP, many researchers have tried to relax the MTSP to the canonical TSP and use exact algorithms to solve it, yet the results have always been unsatisfactory (Bektas, 2006). Most of the research related to the use of GAs for the vehicle scheduling problem have focused on using two different chromosome designs (i.e. one chromosome representation and two chromosome representation) for the MTSP. Both of these chromosome designs can be manipulated using classic GA operators developed for the TSP; however, they are also prone to produce redundant solutions to the problem. Carter and Ragsdale (2006) proposed a GA for the MTSP that uses a two-part chromosome representation which they proved could effectively reduce such redundant solutions in the search space. The crossover operation for the two-part chromosome is separated into two independent operations. The first operation uses an ordered crossover operator, while the second operation uses an asexual crossover operator to ensure that the second part of the chromosome remains feasible. However, due to the nature of the two-part chromosome, Carter (2003) has suggested that further research into "more effective crossover operators" would be both important and necessary. To our knowledge, no specific crossover operator

* Corresponding author. Tel.: +61 295143142; fax: +61 295142655.
E-mail addresses: marshall.s.yuan@gmail.com (S. Yuan), Brad.Skinner@uts.edu.au (B. Skinner), Shoudong.Huang@uts.edu.au (S. Huang), Dikai.Liu@uts.edu.au (D. Liu).

0377-2217/$ - see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.ejor.2013.01.043

for the two-part chromosome in GAs has so far been proposed for solving MTSP. In this paper, we adapt the two-part chromosome representation method to encode the solution for the MTSP, and introduce a new crossover operator which dramatically improves the solution quality.

The paper is organised as follows. Section 2 provides a brief overview of the related work on solving the MTSP using GAs. Section 3 introduces the existing two-part chromosome encoding approach and associated crossover operation. The limitations of the existing method are presented in Section 4. In Section 5, we introduce a new crossover operator for two-part chromosomes and mathematically analyse the advantages of the proposed approach. Section 6 introduces the testing methodology and Section 7 presents the experimental results. Finally, Section 8 provides a conclusion and summary of the study.

2. Literature review

Although the TSP has received a great deal of attention, the research on the MTSP is relatively limited and most of the work is related to MTSP applications. Apart from GAs, other bio-inspired optimisation algorithms such as ant colony optimization (ACO), artificial neural network (ANN) and particle swarm optimization (PSO) have been used to solve the TSP/MTSP (Kulkarni and Tai, 2010). These algorithms are used in conjunction with various local improvement/heuristic techniques to help avoid local minima and also to reduce the computational cost. For example, Liu et al. (2009) proposed an ACO algorithm for solving the MTSP. In their algorithm, the pheromone trail updating and limits followed the MAX–MIN Ant System scheme, and a local search procedure was used to improve the performance of the algorithm. The authors compared the results of the algorithm with existing GA approaches on benchmark instances in the literature. Computational results show that their algorithm was competitive over two typical objective functions.

Some fundamental ways of solving the MTSP are expansion of the problem by converting it into the standard TSP and the simplification of the problem by using the approach of cluster-first-route-second. The work in Bellmore and Hong (1974) is one of the earliest to expand the MTSP by converting it into the standard TSP by introducing $m-1$ imaginary bases/depots. This makes the total number of cities $n+m-1$. This increases the problem size to almost double (Bellmore and Hong, 1974; Kulkarni and Tai, 2010). It also extends the cost matrix to $(n+m-1)\times(n+m-1)$ (Christofides et al., 1981; Kulkarni and Tai, 2010), making the problem computationally tedious compared to the standard TSP with the same number of cities. This becomes worse when the number of cities is very large.

In recent work, Xu et al. (2011) analysed an extended Christofides heuristic for the k-depot TSP (multiple depot multiple travelling salesman problem or k-MDMTSP) and provided proof showing that it achieves a tight approximation ratio of $(2-1/k)$, which is better than the 2-approximation algorithm available in the current literature, and is close to 3/2 when k is close to 2. The proof of this is based on the derivation of new upper bounds for the minimum perfect matching used in the extended heuristic.

We acknowledge that GA is not the only technique for solving the MTSP as presented in Bektas (2006). However, the purpose of this paper is not to focus on benchmarking our current GA implementation against different methods, rather to extend the existing crossover techniques used for the 2-part chromosome approach and examine its efficacy and performance against existing crossover methods for solving the MTSP. Moreover, the TCX operator and genetic algorithm presented in this paper are derived from the class of meta-heuristic algorithms used to find near-optimal solutions to the MTSP problem.

Carter and Ragsdale (2006) proposed a new GA chromosome and related operators for the MTSP and compared the theoretical properties and computational performance of the proposed technique to previous research. Their computational testing shows that the two-part chromosome encoding approach results in a smaller search space and, in many cases, produces better solutions than previous techniques. Most importantly, this work minimises the number of redundant candidate solutions in the search space. Singh and Baghel (2009) proposed a new grouping GA based approach for the MTSP and compared their results with other approaches available in the literature. Their results showed that their approach outperformed the others on the same two objectives, particularly for the computation time. However, in these two approaches (Carter and Ragsdale, 2006; Singh and Baghel, 2009), a local search heuristic is employed to seed the GA with a very good starting point.

In general, GAs are global optimisation techniques (Bo et al., 2006; Cus and Balic, 2003; Goldberg, 1989; Holland, 1992; Smith and Smith, 2002) that avoid many of the shortcomings that exist in classical local search techniques on difficult search spaces. In particular, crossover is a very powerful mechanism for introducing new genetic material and maintaining genetic diversity, but with the defining property that good parents also produce well-performing or superior offspring. Although hybridising the GA with a local optimisation technique to create a memetic algorithm can speed up the overall search process, it is difficult to evaluate the effectiveness of the particular crossover method employed by the GA. We therefore focus in this paper on crossover operators without any local optimisation techniques.

Generally, one-point crossover and two-point crossover are used as two basic genetic crossover operators for many problems (Chatterjee et al., 1996; Kellegöz et al., 2008). One-point crossover is the most basic crossover operator proposed by Holland (1975), and the specific operation process is implemented thus: stochastically select a crossover point in the individual string, exchange the genes before and after the crossover point(s), then two new individuals are generated. One-point crossover tends to treat some loci preferentially as the segments exchanged between two parental chromosomes always contain the endpoints of the gene string. To reduce such positional bias, two-point crossover can be used. Two-point crossover (Holland, 1975) can also be viewed as an improvement on one-point crossover. The difference between them is that two-point crossover chooses two points in the two individual strings which are mated with each other, and then exchanges the gene segments between the selected two crossover points. In an early study, Spears and DeJong (1991) found that two-point and in particular multi-point crossover encourages exploration of the search space due to the increase in gene disruption. This helps to prevent premature convergence to highly fit individuals, making the search more robust.

Many attempts have been made to discover an appropriate crossover operator for TSP, and the most widely used crossover operators are: the ordered crossover operator (ORX), the cycle crossover operator (CYX) and the partially-matched crossover operator (PMX). The ORX, proposed by Davis (1985) and Gen and Cheng (1997) constructs an offspring by choosing a sub-sequence of one parent and preserving the relative order of cities of the other parent. This crossover operator has been commonly used in GAs for solving many problems, particularly for TSP and MTSP (Bektas, 2006; Kulkarni and Tai, 2010). The CYX operator was proposed by Oliver et al. (1987) and attempts to create an offspring from the parents where every position is occupied by a corresponding element from one of the parents. The PMX operator, suggested by Goldberg and Lingle (1985) builds an offspring by choosing a sub-sequence of a chromosome from one parent and preserving the order and position of as many genes as possible from the other

Please cite this article in press as: Yuan, S., et al. A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2013.01.043

parent. However, these crossover methods cannot be directly applied in the GA with the two-part chromosome representation. Carter and Ragsdale (2006) proposed a combined crossover approach ORX + A (ORX combined with an asexual crossover (Chatterjee et al., 1996)) for the two-part chromosome representation. They employed the ORX for the first part of the chromosome and used an asexual crossover for the second part of the chromosome. Section 3 provides an overview of the existing approach. In this paper, we propose a new crossover approach for the MTSP using the two-part chromosome representation, and compare the performance with the three crossover methods (ORX + A, CYX + A and PMX + A) for the two-part chromosome.

3. Overview of the two-part chromosome representation and the existing crossover method

The key to finding a good solution using GAs lies in developing a good chromosome representation of candidate solutions to the problem. Ideally, a good chromosome representation should reduce or eliminate redundant chromosomes from the GA population. Redundancy in the chromosome representation refers to a solution that is capable of being represented in more than one way and appearing in the population multiple times. These multiple representations increase the search space unnecessarily and inhibit the search process. This section presents an overview of the two-part chromosome representation (Carter and Ragsdale, 2006) and the existing crossover method.

3.1. The two-part chromosome encoding technique for solving the MTSP

The two-part chromosome technique, as the name implies, divides the chromosome into two parts. The first part of length n represents a permutation of n cities and the second part of length m gives the number of cities assigned to each salesman. Therefore, the total length of the chromosome is $n + m$ in this representation. The m values present in the second part of the chromosome must sum to n in order to represent a valid solution. In the example of Fig. 1, the first salesman visits cities in the order of 6, 9, 1 and 7, the second salesman visits cities in the order of 8 and 4, and the third salesman travels to cities in the sequence of 5, 2 and 3.

Using the two-part chromosome for solution representation, there are $n!$ possible permutations for the first part of the chromosome. The second part of the chromosome represents a positive vector of integers $(x_1, x_2, \ldots, x_m)$ satisfying $x_1 + x_2 + \cdots + x_m = n$ where $x_i > 0$, $i = 1, 2, \ldots, m$. There are $\binom{n-1}{m-1}$ distinct positive integer-valued m *vectors* that satisfy this requirement. Hence, the solution space for the two-part chromosome is of size $n!\binom{n-1}{m-1}$ (Carter and Ragsdale, 2006).

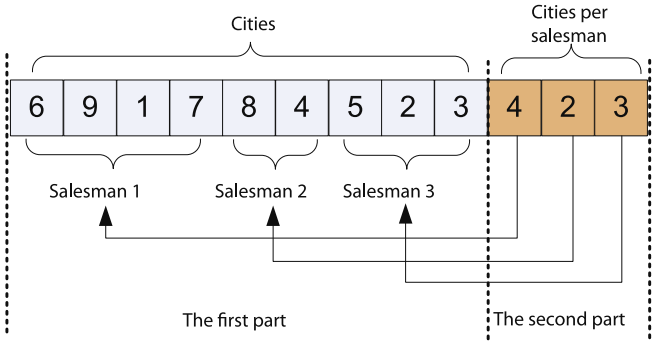


Fig. 1. Example of the two-part chromosome encoding for a 9 city MTSP with 3 salesmen.

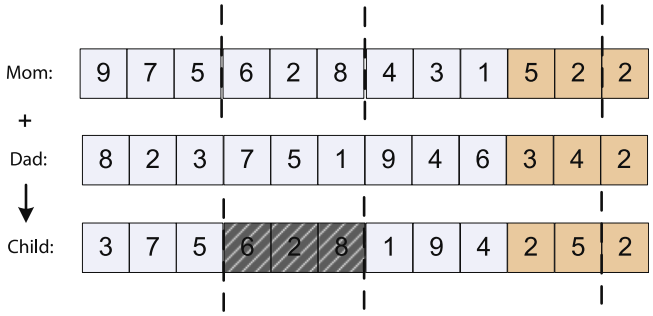


Fig. 2. Example of using the crossover method (Carter and Ragsdale, 2006) for two-part chromosomes.

In addition, two other, different chromosome representations are commonly used when solving the MTSP using GAs; they are the one-chromosome representation (Tang et al., 2000) and two-chromosome representation (Malmborg, 1996; Park, 2001). However, both of these chromosome encoding schemes have a larger number of redundant solutions in the search space compared to the two-part chromosome representation (Carter and Ragsdale, 2006). The size of the solution space for the one-chromosome representation is $(n + m - 1)!$ (Carter and Ragsdale, 2006) and the solution space for the two-chromosome representation is $n!m^n$ (Carter and Ragsdale, 2006).

3.2. Existing crossover method for two-part chromosomes

(Carter and Ragsdale, 2006) employed a combined crossover technique for two-part chromosomes. The first part of the chromosome uses the classic ordered crossover operator (Goldberg, 1989), while the second part of the chromosome uses an asexual crossover operator (Chatterjee et al., 1996). That is, when performing the crossover operation, each portion of the two-part chromosome is treated separately and is processed using two independent crossover methods.

Specifically, Fig. 2 is an example illustrating the overall process of the crossover operation for the canonical two-part chromosome. This example shows that two parent chromosomes generate a child based on Mom's genes. For the first part of the chromosome, the ordered crossover behaves in the following way: the Child inherits its middle section from Mom, and its left and right sections are determined according to the Dad's genes. The shadowed genes (6, 2, 8) in the first part of the Child's chromosome are directly copied from Mom, and other genes are filled according to the sequence of remaining genes in Dad's chromosome. For the second part of the chromosome, a single point asexual crossover operator is utilised as shown in Fig. 2. This method simply cuts the second part of Mom's chromosome into two sections ((5, 2) and (2)) and reverses the order in which the two gene segments were arranged. This type of crossover ensures that the second part of the chromosome remains feasible (with the sum of the values in the chromosome equalling n). Likewise, when performing crossover based on Dad's genes, another child can be generated as well. Normally, each pair of parents can generate two offspring.

4. Limitations of the existing method

The crossover method in Carter and Ragsdale (2006) has two limitations, which can reduce the search performance. First, in order to ensure that the second part of the chromosome is valid (i.e. a

Please cite this article in press as: Yuan, S., et al. A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2013.01.043

Table 1
The number of fundamental combinations in the second part of the chromosome for each MTSP problem (number of cities_number of salesmen) showing the exponentially increasing number of combinations with increasing m.

51_3	51_5	51_10	100_3	100_5	100_10	100_20	150_3	150_5	150_10	150_20	150_30
2.17E+02	2.82E+03	1.95E+04	8.33E+02	3.82E+04	2.98E+06	1.05E+07	1.88E+03	1.88E+05	–	–	–

Table 2
Limited search space of the existing method for the second part of the chromosome.

51_3	51_5	51_10	100_3	100_5	100_10	100_20	150_3	150_5	150_10	150_20	150_30
46.08%	3.55%	0.51%	12.00%	0.26%	0.00%	0.00%	5.33%	0.05%	0.00%	0.00%	0.00%

Table 3
Computational test conditions.

Number of cities (n)	Number of salesmen (m)	GA generations
51	3, 5 and 10	50,000
100	3, 5, 10 and 20	100,000
150	3, 5, 10, 20 and 30	200,000
128	10, 15 and 30	200,000

Table 4
Parametric configuration for GA.

Parameter	Value
GA population size	100
Crossover probability rate in GA	0.85
Mutation probability rate in GA	0.01
Selection method in GA	*Roulette-wheel selection*
Mutation method	*Swap*
Replacement in GA	*Steady-state GA*
Replacement percentage	20%

positive vector of integers $(x_1, x_2, \ldots, x_m)$ satisfying $x_1 + x_2 + \cdots + x_m = n$ where $x_i > 0$, $i = 1, 2, \ldots, m$), an asexual crossover has been adopted. The asexual crossover operator only changes the order of genes in the second part of the chromosome. This approach may limit the diversity of the whole population, because some other feasible numerical combinations would not appear in the second part of the chromosome, during the initialisation of the population. For example, in Fig. 2, the second part of Mom and Dad's chromosomes are $\langle 5, 2, 2\rangle$ and $\langle 3, 4, 2\rangle$ respectively and their children would never have a chance to reach other feasible genes in the second part of their chromosomes, such as $\langle 1, 1, 7\rangle$, $\langle 1, 2, 6\rangle$ and $\langle 1, 3, 5\rangle$, which are all still valid and feasible chromosomes. Additionally, for the MTSP problem ($n = 9, m = 3$), there are 7 fundamental gene combinations: $\langle 1, 1, 7\rangle$, $\langle 1, 2, 6\rangle$, $\langle 1, 3, 5\rangle$, $\langle 1, 4, 4\rangle$, $\langle 2, 2, 5\rangle$, $\langle 2, 3, 4\rangle$ and $\langle 3, 3, 3\rangle$. Therefore, the GA population must contain the 7 fundamental gene combinations in the second part of the chromosome, so that the asexual crossover may have the chance to reach all possible solutions in the second part of the chromosome.

When the size of the MTSP problem gets bigger, the number of fundamental combinations increases dramatically and it is impossible to have all possible combinations in the limited population. To calculate the number of fundamental combinations in the second part of the chromosome, we define a function $f(n, m, k)$, where n is the number of cites, m is the number of salesmen; k is the minimum number of cities each salesman must visit. In our case, we only consider $f(n, m, 1)$, where $k = 1$ and each salesman must visit at least one city. However, the general form $f(n, m, k)$ is necessary for a recursion process to make the calculation relatively convenient. The recursive function can be defined as:

$$f(n,m,k) = \sum_{i=k}^{\lfloor \frac{n}{m} \rfloor} f(n-i, m-1, i) \tag{1}$$

where $\lfloor \frac{n}{m} \rfloor$ is the floor value of (n/m), which means that each salesman has to visit at least $\lfloor \frac{n}{m} \rfloor$ cities. When $m = 1$, $f(n,1,k) = 1$. That is, if there is only one salesman, then the number of fundamental combinations is only one (i.e. $\langle n\rangle$), regardless of the values of n and k.

Proof of Eq. (1): Suppose the number of cities visited by each of the m salesman is represented by $(x_1, x_2, \ldots, x_m)$. To compute the number of fundamental combinations we just need to consider the non-decreasing combinations, that is $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_m$. Since the total number of cities is n and the total number of salesmen is m it is clear that the upper limit of x_1 is $\lfloor \frac{n}{m} \rfloor$, otherwise there must be some $x_i < x_1$ which is not valid. Now since we are computing $f(n, m, k)$ and each salesman needs to visit at least k cities, the minimal value of x_1 is k. Thus all the possible values for x_1 are from k to $\lfloor \frac{n}{m} \rfloor$ and why in Eq. (1) the sum of i is from k to $\lfloor \frac{n}{m} \rfloor$. Now suppose x_1 is fixed as i (any integer from k to $\lfloor \frac{n}{m} \rfloor$), then the problem is for $m - 1$ salesmen to visit $(n - i)$ cities. Since we

Table 5
Experimental results for minimising total travel distance.

Problem	Crossover	$m = 3$			$m = 5$			$m = 10$			$m = 20$			$m = 30$		
		Mean	Stdev	Best	Mean	Stdev	Best	Mean	Stdev	Best	Mean	Stdev	Best	Mean	Stdev	Best
MTSP-51	TCX	510	24	466	536	26	499	636	17	602	–	–	–	–	–	–
	ORX + A	584	29	517	621	39	551	709	33	648	–	–	–	–	–	–
	CYX + A	591	43	511	622	44	530	710	42	633	–	–	–	–	–	–
	PMX + A	601	38	513	606	40	537	705	34	625	–	–	–	–	–	–
MTSP-100	TCX	32,708	2267	28,943	34,179	2006	30,941	36,921	1964	32,802	46,976	1773	44,112	–	–	–
	ORX + A	41,516	3356	36,713	42,416	2806	36,196	44,631	2997	38,717	54,265	3059	47,971	–	–	–
	CYX + A	41,911	3195	35,791	43,634	2804	35,421	45,150	3241	40,894	52,916	2884	46,466	–	–	–
	PMX + A	41,441	3423	33,802	42,063	3931	33,908	44,786	3467	39,785	52,142	2688	46,212	–	–	–
MTSP-150	TCX	55,851	2588	51,126	61,596	4759	51,627	61,360	3888	54,473	69,701	4340	62,456	84,008	5285	76,481
	ORX + A	67,037	3745	60,090	68,018	3377	62,539	72,113	3637	63,899	81,696	5372	71,933	96,122	4562	88,515
	CYX + A	67,463	4454	55,335	69,860	4342	61,521	71,584	4845	63,126	83,471	4197	75,146	97,106	3911	89,008
	PMX + A	68,152	5140	58,303	69,112	4011	60,761	72,620	4334	64,975	81,178	4920	73,281	95,752	4923	87,402

Please cite this article in press as: Yuan, S., et al. A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2013.01.043

are only considering non-decreasing combinations, the minimal number of cities that each salesman needs to visit is $x_1 = i$, giving $f(n-i, m-1, i)$ in Eq. (1).

For example, given $n = 5$, $m = 2$ and $k = 1$ then the total number of fundamental combinations can be calculated using Eq. (1):

$$\begin{aligned} f(5,2,1) &= \sum_{i=1}^{2} f(5-i, 2-1, i) \\ &= f(5-1, 2-1, 1) + f(5-2, 2-1, 2) \\ &= f(4,1,1) + f(3,1,2) = 1+1 = 2 \end{aligned}$$

In fact, there are only two fundamental combinations $\langle 1, 4\rangle$ and $\langle 2, 3\rangle$ for the case $n = 5$ and $m = 2$. Based on Eq. (1), we provide an analysis of the number of fundamental combinations in the second part of the chromosome for a MTSP by increasing the problem size. Table 1 shows the exponential nature of the possible number of fundamental combinations for 12 MTSP problems (Carter and Ragsdale, 2006). Note that the cell with "–" means that the number of combinations is greater than (7E+07). In addition, Table 2 shows the limited number of fundamental combinations (as a percentage) when using the existing crossover method, compared to the actual numbers of all fundamental combinations. Assume that the initial population size is 100, and all second part chromosomes contain different fundamental combinations. Then the percentages are calculated using the population size (100) divided by the number of calculated fundamental combinations in Table 1. Obviously, the limitation of using the existing crossover approach is that it does not have the opportunity to reach more fundamental combinations as the initial population size is limited.

Another important characteristic of GAs is that children are expected to stochastically inherit a subset of the parents' genes during each crossover process. For the MTSP, each salesman has an independent tour, which represents the sequence of visited cities by the salesman himself and does not have any direct relationship with other salesmen in the MTSP. However, when using the existing crossover method, the likelihood of children inheriting parents' sub-tours is relatively limited. This is mainly because the existing crossover method treats the first part of the chromosome as a whole and ignores the important mapping of gene segments to independent salesmen in the second part of the chromosome. In other words, the existing method would potentially break important building blocks or highly fit gene segments (tours) of the parent chromosomes. This situation may worsen when the number of salesmen (m) and the number of cities (n) increase.

Overall, the search performance of GAs with the two-part chromosome representation can be seriously degraded as a result of the above two shortcomings.

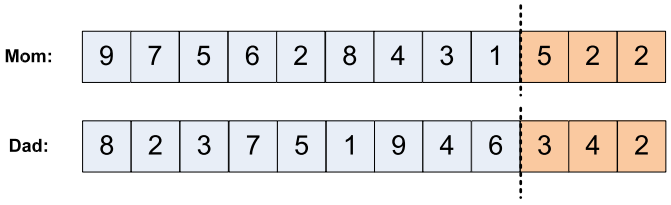
5. A new crossover approach

In this section, we propose a new crossover approach (TCX) for the two-part chromosome crossover to improve the GA search performance for solving the MTSP.

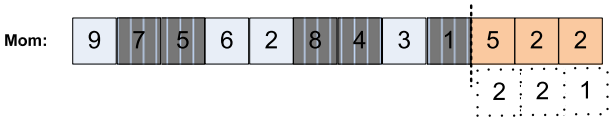
TCX treats each salesman separately when performing crossover in the first part of the chromosome. This ensures that highly fit building blocks that may be present in the subtours of parental chromosomes are maintained during the reproduction process and inherited by the offspring chromosomes. In addition, TCX greatly enhances the diversity in the second part of the chromosome by increasing the number of feasible combinations, while always satisfying the constraint: $(x_1, x_2, \ldots, x_m)\ \ x_1 + x_2 + \cdots + x_m = n$ where $x_i > 0$, $i = 1, 2, \ldots, m$. Instead of using the existing asexual crossover, TCX reproduces genes in the second part of the chromosome according to the results of the crossover operation that was performed in the first part of the chromosome. Therefore, new types of fundamental combinations may be generated and the diversity in the second part of the chromosome is increased dramatically.

An example is used below to illustrate the process of using TCX to generate a child chromosome of (9 cities and 3 salesmen). Five basic steps involved. Firstly, a pair of parent two-part chromosomes (Mom and Dad) are initialised, and the Mom chromosome is used as the base for generating a Child in this example. Secondly, we randomly select a gene segment (i.e. sub-tour) from the first part of Mom's chromosome. In this case, the selected gene segments are $\langle 7, 5\rangle$ for salesman 1, $\langle 8, 4\rangle$ for salesman 2 and $\langle 1\rangle$ for salesman 3. Hence, the numbers of randomly selected genes are (2, 2, 1) as shown in the dashed area of Step 2. Next, the order of the remaining genes is sorted according to the positions in the first part of Dad's chromosome. In this example, the remaining genes in the first part of Mom's chromosome are $\langle 9, 6,\ 2, 3\rangle$, and the order of these genes is shuffled to $\langle 2, 3, 9, 6\rangle$ according to the first part of Dad's chromosome. Next, based on the number of unselected genes, we compute a uniform random number, between 1 and the current value of unselected genes, to determine how many new genes will be added for each salesman in the child chromosome. Here, the unselected genes are $\langle 2, 3, 9, 6\rangle$ and if, for example, we iteratively generate the integer sequence (2, 1, 1) of uniform random numbers between 1 and the number of currently unselected genes, then in the Child chromosome, salesman 1 gets genes $\langle 2, 3\rangle$, salesman 2 gets $\langle 9\rangle$, and salesman 3 gets $\langle 6\rangle$. Moreover, during this step it is possible to generate a sequence containing one or more zeros, for example (4, 0, 0) or (3, 1, 0). This does not create infeasibility in the second part of the chromosome. Therefore, for the first part of the Child's chromosome, salesman 1 would have $\langle 7, 5, 2, 3\rangle$, salesman 2 would have $\langle 8, 4, 9\rangle$, and salesman 3 would have $\langle 1, 6\rangle$. Lastly, we construct the two-part chromosome for the Child by updating the information in the second part of its chromosome. In this example, $\langle 4, 3, 2\rangle$ is generated by summing the each position of the intermediary second part vectors of $\langle 2, 2, 1\rangle + \langle 2, 1, 1\rangle = \langle 4, 3, 2\rangle$, which remains feasible and bounded.

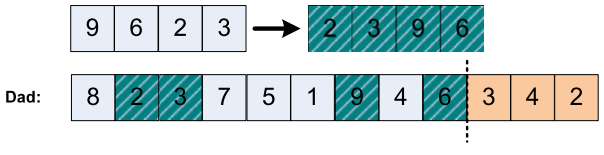
Step 1: Initialise a pair of chromosomes as parents



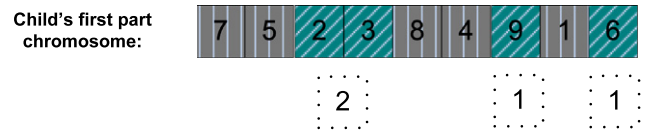
Step 2: Randomly select a gene segment for each salesman



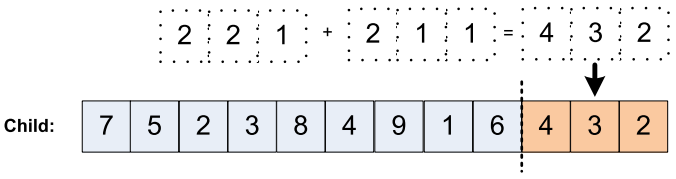
Step 3: Shuffle gene positions according to the first part of Dad's chromosome



Step 4: Add genes for each salesman



Step 5 : Construct the Child's two-part chromosome



In the second part of the Child's chromosome, the TCX operator will generate a new combination, which is probably different from the original fundamental combination in the second part of Mom's chromosome. As shown in the above example, the second part of the Child's chromosome is $\langle 4, 3, 2\rangle$, while the Mom's is $\langle 5, 2, 2\rangle$. However, if using the existing crossover approach, the second part of the Child's chromosome would have been the same as the Mom's chromosome $\langle 5, 2, 2\rangle$. Subsequently, taking Dad's chromosome as the base, another Child chromosome can be produced by going through the five steps described above. A pair of Child chromosomes is generated after each TCX crossover operation. The following pseudo code shows the proposed TCX crossover method.

Two-part chromosome crossover operation:
Input: A pair of two-part parent chromosomes.
Output: A pair of two-part child chromosomes.
//Process of generating a Child (Sister)
Select a parent's chromosome as the Mom base for the Child;
for $m = 1$:*numSalesmen*//*numSalesmen: the total number of salesmen{*
Randomly generate an integer number between 1 and *AssignedCities[m]* and store in *Segment[m]*;//
assignedCities[m]: the number of assigned cities of salesman m
if *AssignedCities[m]* > *Segment[m]*
Randomly generate an integer number between 1 and (*AssignedCities[m]* − *Segment[m]*) and set it as the possible starting position for the gene segment;
for $k = 1$:*Segment[m]*
Copy each gene from the gene segment for a Child;
Copy each gene into *savedGenesPool*;
end
else
Copy the entire part of salesman m for Child;
Copy the entire part of salesman into *savedGenesPool*;
end
totalSavedGenes = *totalSavedGenes* + *Segment[m]*;
end
totalUnsavedGenes = *numCities* − *totalSavedGenes;*
for $m = 1$:*numSalesmen*
if $m! = numSalesmen$
Randomly generate an integer number between 1 and *totalUnsavedGenes* for salesman m to add genes;
else
The salesman will add all unsaved genes;
end
According to the order of the unsaved genes in the first part of Dad's chromosome, add the randomly generated number of genes to the Child;
end
Based on the construction of the first part of the Sister's chromosome, calculate the number of assigned cities in the second part of the chromosome.
//Process of generating a Child (Brother)
Select another parent's chromosome as the base for the Child;
Repeat the process of generating the Sister;

It can be seen that, TCX has two advantages when two-part chromosomes are used to solve MTSP problems. Firstly, the offspring have a better chance of inheriting highly-fit subtours from the parental chromosomes, because the mapping between each salesman and the corresponding MTSP subtour are considered separately in TCX. The second advantage is the dramatic increase in diversity of the second part of the chromosome because children have the opportunity to reach any feasible combinations within the entire search space defined in the second part. The experimental performance of TCX and a comparison with the other three crossover methods will be presented in Section 7.

6. Computational testing methodology

To evaluate the benefits of the proposed TCX crossover operator, computational experiments were conducted to compare the performance of four crossover methods on a set of problems created for the MTSP We compare the proposed TCX to ORX + A, CYX + A and PMX + A, which are combined with an asexual crossover for the two-part chromosome representation. The test problems were selected from a standard collection of TSPs from the Library of Travelling Salesman Problems (Reinelt, 2001) that were transformed into MTSPs by using more than one salesperson (m) to complete the tour. The test problems are Euclidean, two-dimensional symmetric problems with 51, 100, and 150 cities. Throughout this paper, we refer to these test problems as MTSP-51, MTSP-100, and MTSP-150, respectively. These MTSP problems appear to be well-suited to the purpose of this paper, which is to examine the limitations of the existing crossover operators for 2-part chromosomes and to compute the effectiveness of the proposed TCX operator. They permit direct comparison with the previous approaches and even though they are a simplified representation of practical problems, they remain representative as benchmarking problems.

Two different fitness (or objective) functions were considered. The first fitness function measured the total travel distance travelled by all of the salesmen. For the problems, each of the m salesmen were required to visit at least one city (other than the home city). This objective represents a situation in which there is a group of salesmen and there are no constraints associated with the maximum number of cities visited by any one salesman. The second fitness function measured the longest route among the m salesmen. Typically, this objective function is used to equalise the workload among the available salesmen when scheduling tasks.

In our simulation experiments, all GA programs were implemented in C++ with GAlib 2.4.7 (GAlib, 2007). All experiments were performed on a High Performance Computing Linux Cluster, configured using 2×3.33 Ghz Quad Core Xeon with 24 GB of DDR3-1333 ECC Memory (the hardware specification can be found at: https://clusterportal.feit.uts.edu.au/pages/about/hardware).

Please cite this article in press as: Yuan, S., et al. A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2013.01.043

Table 6
t-Test results of TCX versus (ORX, CYX and PMX) crossover approaches for minimising total travel distance.

Problem	Comparison	$m = 3$	$m = 5$	$m = 10$	$m = 20$	$m = 30$
MTSP-51	TCX VS ORX + A	−10.77	−10.03	−10.93	–	–
	TCX VS CYX + A	−8.96	−9.32	−9.13	–	–
	TCX VS PMX + A	−11.04	−8.16	−9.98	–	–
MTSP-100	TCX VS ORX + A	−11.91	−13.08	−11.78	−11.29	–
	TCX VS CYX + A	−12.87	−15.02	−11.89	−9.61	–
	TCX VS PMX + A	−11.65	−9.79	−10.81	−8.79	–
MTSP-150	TCX VS ORX + A	−13.46	−6.03	−11.06	−9.51	−9.50
	TCX VS CYX + A	−12.35	−7.03	−9.01	−12.49	−10.91
	TCX VS PMX + A	−11.71	−6.61	−10.59	−9.58	−8.91

Table 7
Comparison of the mean results between two approaches using GA with seeding enabled for minimising total travel distance.

Problem	m	Carter and Ragsdale (2006)	TCX	Improvement (%)
MTSP-51	3	543	492	9.39
	5	586	519	11.43
	10	723	670	7.33
MTSP-100	3	26,653	26,130	1.96
	5	30,408	28,612	5.91
	10	31,227	30,988	0.77
	20	54,700	44,686	18.31
MTSP-150	3	47,418	44,674	5.79
	5	49,947	47,811	4.28
	10	54,958	51,326	6.61
	20	73,934	62,400	15.60
	30	99,547	78,023	21.62

6.1. GA issues

Fundamentally, GAs use operators such as selection, crossover, mutation and replacement as means of preserving beneficial information with the overall goal of finding a better solution to the problem. In addition, GAs work using codification of the parameter space rather than the parameters themselves. The objective function can be easily defined as a measure of fitness for solution performance, which allows the GA to retain useful solutions and inhibit those which are less useful. With the two-part chromosome representation, classic crossover methods such as ORX, CYX and PMX cannot be directly applied in the GA to solve the MTSP, because infeasible solutions would be created as a result of destroying the two-part chromosome structure. Hence, in this paper, we use the proposed TCX for the two objective functions, and compare the results with ORX + A, CYX + A and PMX + A.

Table 9
t-Test results of TCX versus (ORX, CYX and PMX) crossover approaches for minimising longest tour.

Problem	Comparison	$m = 3$	$m = 5$	$m = 10$	$m = 20$	$m = 30$
MTSP-51	TCX VS ORX + A	−2.91	−4.41	−6.33	–	–
	TCX VS CYX + A	−4.05	−3.06	−6.27	–	–
	TCX VS PMX + A	−3.71	−3.26	−7.79	–	–
MTSP-100	TCX VS ORX + A	−2.38	−6.01	−5.66	−7.56	–
	TCX VS CYX + A	−4.76	−4.73	−5.22	−7.12	–
	TCX VS PMX + A	−2.80	−4.63	−6.39	−8.45	–
MTSP-150	TCX VS ORX + A	−5.88	−4.47	−10.98	−11.15	−11.83
	TCX VS CYX + A	−3.30	−4.28	−11.11	−12.06	−15.55
	TCX VS PMX + A	−4.25	−5.31	−8.85	−14.90	−11.39

Table 10
Comparison of crossovers with GA seeding for minimising longest tour.

Problem	m	Carter and Ragsdale (2006)	TCX	Improvement (%)
MTSP-51	3	203	203	0.00
	5	164	154	6.10
	10	123	113	8.13
MTSP-100	3	13,556	12,726	6.12
	5	10,589	10,086	4.75
	10	9463	7064	25.35
	20	8388	6402	23.68
MTSP-150	3	19,687	18,019	8.47
	5	14,748	12,619	14.44
	10	11,158	8054	27.82
	20	10,044	5673	43.52
	30	8775	5270	39.94

6.1.1. Population initialisation

We employ the same method as Carter and Ragsdale (2006) to generate the initial population. The first part of each chromosome is a randomly generated permutation of the n cities. As the sum of the m positive integers in the second part of the chromosome must equal n, m gene values (x_i) for the second part of the chromosome are generated as a discrete uniform random number between 1 and $n - \sum_{k=1}^{i-1} x_k$, for $i = 1$ to m (i.e., the maximum value of each

Table 8
Experimental results for minimising longest tour.

Problem	Crossover	$m = 3$			$m = 5$			$m = 10$			$m = 20$			$m = 30$		
		Mean	Stdev	Best	Mean	Stdev	Best	Mean	Stdev	Best	Mean	Stdev	Best	Mean	Stdev	Best
MTSP-51	TCX	207	13	182	153	10	135	113	2	112	–	–	–	–	–	–
	ORX + A	216	12	191	165	11	139	128	13	112	–	–	–	–	–	–
	CYX + A	222	16	188	161	12	138	131	16	112	–	–	–	–	–	–
	PMX + A	218	11	191	161	10	141	130	12	112	–	–	–	–	–	–
MTSP-100	TCX	14,365	1013	12,645	10,086	674	8730	7768	492	6796	6768	433	6358	–	–	–
	ORX + A	15,137	1462	12,997	11,459	1053	9415	9286	1385	7373	8123	881	6666	–	–	–
	CYX + A	15,759	1242	13,467	11,333	1278	9507	9151	1364	7111	8109	936	6516	–	–	–
	PMX + A	15,238	1371	12,249	11,233	1177	9267	6890	1482	7187	8265	868	6570	–	–	–
MTSP-150	TCX	22,523	1226	20,556	16,054	1227	14,096	10,722	927	8475	9640	789	8423	8759	806	7169
	ORX + A	24,766	1689	22,015	17,646	1519	15,266	15,150	2006	11,788	13,669	1816	10,274	12,204	1376	9182
	CYX + A	23,906	1941	20,915	17,608	1563	14,029	14,738	1750	10,779	14,111	1872	8365	13,091	1295	10,694
	PMX + A	24,216	1802	20,347	17,741	1235	15,418	14,489	2139	10,738	13,810	1315	11,722	12,608	1667	10,080

Please cite this article in press as: Yuan, S., et al. A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2013.01.043

successive gene value was based on n and the sum of the previous values). These values are then randomly assigned to the genes in the second part of the chromosome.

6.1.2. Selection

We utilise the rank-based roulette-wheel (Goldberg, 1989) selection in our GA. With this approach, each chromosome is assigned a portion of an imaginary roulette wheel, based on a chromosome's rank. Chromosomes which have a higher fitness value are allocated a larger segment of the roulette wheel. Selection of a parent requires random generation of a number between 0 and 1. The chromosome occupying the section of the roulette wheel covered by the randomly generated percentage is chosen as a parent. The second parent is selected in the same manner.

6.1.3. Mutation

The mutation operator used in our GA is based on the swap mutation (Goldberg, 1989), which consists of randomly swapping two genes in both parts of the two-part chromosome. The swap mutation operator performs a vital task because it provides the means by which jobs are exchanged to seek improvement. Crossover has the ability to drastically impact the groupings of jobs, but has little impact on the ordering of the jobs within each group. Mutation is needed to explore alternative solutions, because a low probability of mutation can maintain diversity of the solutions in the population. The swap mutation operation is applied to the first-part and second-part of the chromosome through independent application of the mutation operator.

6.1.4. Replacement

We use the replacement policy of Steady-State GA (DeJong, 1975). Parents are selected to produce offspring and then a decision is made which individuals in the population to select for deletion to make room for the new offspring. Steady-State GA is an overlapping population policy, since parents and offspring compete for survival. A percentage of the replacement determines how much of the population is replaced by the newly generated children in each generation.

6.1.5. Experimental parameters

Table 3 summarises the experimental conditions of 12 different problem size (n) and salesman (m) combinations along with the run times for each type of problem. The stopping criterion is the number of generations. All salesmen start and end their individual tours in the same city. Table 4 shows the parameter values for 'all' the experiments performed in this paper.

7. Computational results

This section presents performance comparisons between the proposed TCX approach and the other three crossover methods for two-part chromosomes. To correctly show the performance differences among the various crossover operators, we conduct two sets of experiments for two objective functions. For each objective function, we first run the GA without seeding (i.e. without any local optimisation) for the benchmark problems and all initial populations are randomly created. Subsequently, we conduct the same benchmark experiments using GA seeding. In this case, each of the starting populations is seeded with a candidate solution produced by a simple greedy heuristic to give the GA a good starting point in the search space.

Since GAs belong to the class of stochastic search algorithms, we employ statistical hypothesis testing using the two-sampled pooled t-test (Walpole et al., 2006). The experiments for the four crossover approaches in this paper are conducted using 30 independent trials (each pair of $n_1 = n_2 = 30$). This occurs for both seeded and unseeded GA experiments. That is, we compare TCX to each of the other three crossover methods. In this paper, the null hypothesis can be stated such that: 'TCX does provide higher solution quality when applied to 30 trials'. Statistical differences are indicated according to the two-sample pooled t-test with a level of significance at $p = 0.05$ (95% confidence). The statistical hypothesis tests in this paper use two-tailed critical regions to indicate whether a significant improvement by TCX ($t \leqslant -2.00$) or significant degradation by TCX ($t \geqslant 2.00$). Statistical tests resulting in a t-value of ($-2.00 < t < 2.00$) do not provide enough statistical evidence to refute or confirm the null hypothesis, which indicates similar performance between the two algorithms. Moreover, we assume that the experimental results follow a standard normal distribution under the null hypothesis and the experimental data has been sampled independently from the two populations being compared. Furthermore we employ a variation of the t-test known as the Welch's t-test which is used when the two population variances are assumed to be different and must be estimated separately. In addition, experimental results and the MTSP problems used in this paper including the cost matrices can be downloaded from http://ims.uts.edu.au/MTSP/MTSP_dataset_solutions.zip.

7.1. Experiments for minimising total travel distance

The first objective function is to minimise the total travel distance of all the salesmen. This objective reflects the goal of minimising the distance required to visit all n cities. The only constraint used with this objective is that each salesman must visit at least one city (other than their home city). Without this constraint, the GA could reduce the number of salesmen in the problem, hence possibly reducing the MTSP to a TSP. For this objective, as the number of salesmen increases the total travel distance of the combined trips also tends to increase, because each salesman must start and return to the home city.

7.1.1. GA without seeding for minimising total travel distance

Table 5 summarises the mean value and standard deviation of the 30 trails of each approach for testing problems with the objective of minimising the total travel distance. The significant improvements in the performance of TCX with respect to each other method are indicated by the t-values. According to the critical value ($t = -2.00$), all calculated t-values as shown in Table 6 are less than -2.00 for the 12 problems. In other words, the solutions found by TCX are statistically significantly better than the solutions found by the other three crossover methods (ORX + A, CYX + A and PMX + A) for all cases (MTSP-51, MTSP-100 and MTPS-150).

7.1.2. GA with seeding for minimising total travel distance

As mentioned previously in the paper, we also test the performance of TCX with GA seeding enabled. We employ the same seeding technique as Carter and Ragsdale (2006) for minimising total travel distance and generate the greedy solutions by examining the present location of all the salesmen and calculating the closest unassigned city to each salesman. The unassigned city is then assigned to the closest salesman and the process is continued until all the cities are assigned to a salesman.

In Table 7, we compare the average results in the literature (Carter and Ragsdale, 2006) with the mean results found by our TCX approach, using the simple GA seeding technique. For the 12 testing problems, TCX shows positive improvements compared to the results obtained in Carter and Ragsdale (2006). The greatest improvement is achieved at MTSP-150 ($m = 30$) with a 21.62% improvement in final solution quality. Therefore, by combining GA seeding and the proposed TCX operator a better solution quality can be achieved when compared to the approach in Carter

Please cite this article in press as: Yuan, S., et al. A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2013.01.043

Table 11
TCX Robustness to varying GA parameters for minimising total travel distance.

	Default		$P = 50$		$P = 200$		$P = 1000$		$Pc = 0.7$		$Pc = 1.0$		$Pm = 0.02$		$Pm = 0.001$	
	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
A	636	17	655	27	623	19	616	18	642	24	645	18	644	20	630	21
B	36,921	1964	37,970	2950	36,072	2119	35,173	1511	38,669	2389	38,872	1844	37,317	2412	36,723	1991
C	61,360	3888	63,565	4221	60,049	4803	56,089	2983	64,411	5079	65,542	3034	88,992	4287	57,798	4259

Table 12
TCX Robustness to varying GA parameters for minimising longest tour.

	Default		$P = 50$		$P = 200$		$P = 1000$		$Pc = 0.7$		$Pc = 1.0$		$Pm = 0.02$		$Pm = 0.001$	
	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
A	113	2	115	4	115	4	112	0	113	3	173	7	113	3	121	6
B	7768	492	8073	562	7560	535	7289	467	7639	556	9224	1277	7610	515	8708	353
C	10,722	927	11,811	1006	10,500	827	9631	850	10,585	879	31,386	885	14,004	1064	12,201	941

Table 13
Optimality gap for small-sized symmetric MTSP problems for minimising total travel distance.

Problem ($m = 3$)	Optimal	TCX_MEAN	TCX_STDEV	TCX_BEST
MTSP-11a	198	198	0	198
MTSP-11b	135	135	1	135
MTSP-12a	199	199	0	199
MTSP-12b	2295	2295	0	2295
MTSP-16	242	247	8	242

Table 14
Optimality gap for small-sized symmetric MTSP problems for minimising longest tour.

Problem ($m = 3$)	Optimal	TCX_MEAN	TCX_STDEV	TCX_BEST
MTSP-11a	77	77	0	77
MTSP-11b	73	73	0	73
MTSP-12a	983	987	4	983
MTSP-12b	77	77	0	77
MTSP-16	94	101	7	94

and Ragsdale (2006) for the objective of minimum total travel distance.

Seeding is not without risk, however, and may cause the GA to become stuck at a local minimum in some cases. For example, the average result 670 for the problem (MTSP-51: $m = 10$) with seeding is worse than the mean result 636 without seeding in Table 5. Clearly, seeding may provide a good start point and reduce the search cost, but it might also reduce solution quality when GA is attracted by a local minimum.

7.2. Experiments for minimising longest tour

Besides the objective of minimising the total travel distance, another common objective in real world MTSP applications is minimising the longest individual tour, which is also called makespan (Yuan et al., 2011). Minimising the longest tour has the goal of balancing the cities (or workload) among the salesmen and minimising the distance the salesmen travel. With the objective of minimising the longest tour, the fitness values decrease as the number of salesmen increase.

7.2.1. GA without seeding for minimising longest tour

Table 8 summarises the mean value and standard deviation of the 30 trials for of each approach for the benchmarking problems for minimising the total travel distance. The significant improvements in performance of TCX with respect to each of the other crossover methods are indicated by the t-values in Table 9. TCX statistically outperforms the other three crossover methods (ORX + A, CYX + A and PMX + A) for the majority of the benchmarking problems ($t < -2.00$).

7.2.2. GA with seeding for minimising longest tour

Likewise, we also tested the performance of TCX with GA seeding enabled. We employ the same seeding technique as Carter and Ragsdale (2006) for minimising longest tour and generate the greedy solutions by iterating through all the salesmen in a round-robin fashion and assigning the closest unassigned city to each salesman in turn. This continues until all the cities are assigned to a salesman.

In Table 10, with the simple GA seeding technique, we compare the average results in Carter and Ragsdale (2006) with the results found by our TCX approach. For the 12 problems, TCX has various positive improvements compared to the results obtained in Carter and Ragsdale (2006). The greatest improvement is achieved at MTSP-150 ($m = 20$) with 43.52% Therefore, by combining GA seeding and the proposed TCX operator better solution quality can be achieved when compared to the approach in Carter and Ragsdale (2006) for the objective of minimum longest tour. Overall, the benchmarking experiments that were conducted for the two objective functions clearly suggest an increase in solution quality for the

Table 15
Comparison of crossover methods for the sgb128 MTSP test problem for minimising the total travel distance.

Problem	Crossover	$m = 10$				$m = 15$				$m = 30$			
		Mean	Stdev	Best	t-Test	Mean	Stdev	Best	t-Test	Mean	Stdev	Best	t-Test
MTSP-128	TCX	42,335	1697	39,478	–	44,294	1911	40,843	–	54,696	2189	50,809	–
	ORX + A	43,958	1214	40,922	−4.33	45,674	2133	40,524	−2.64	56,357	1973	51,439	−3.09
	CYX + A	44,133	1471	40,269	−4.38	45,282	1542	42,366	−2.20	56,376	1911	52,967	−3.17
	PMX + A	44,386	1825	41,671	−4.51	46,159	1926	42,024	−3.76	56,369	1623	53,416	−3.36

Please cite this article in press as: Yuan, S., et al. A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2013.01.043

Table 16
Comparison of crossover methods for the sgb128 MTSP test problem for minimising longest tour.

Problem	Crossover	$m = 10$				$m = 15$				$m = 30$			
		Mean	Stdev	Best	*t*-Test	Mean	Stdev	Best	*t*-Test	Mean	Stdev	Best	*t*-Test
MTSP-128	TCX	6716	428	5912	–	5979	404	5295	–	4814	435	4003	–
	ORX + A	8569	887	6421	−10.31	8166	947	6051	−11.36	6211	475	5356	−11.88
	CYX + A	8464	906	6566	−9.55	8007	739	6324	−13.19	6261	680	4776	−9.82
	PMX + A	8299	848	6880	−9.13	8016	1065	6480	−9.79	6354	632	5540	−11.00

TCX technique, which is further increased through the introduction of initial seeding of the GA population. Taken together, the results presented in this paper suggest that the proposed TCX operator enables the GA to find better solutions for solving the MTSP with the two-part chromosome representation.

7.3. TCX robustness

Since the solution quality and performance of the genetic algorithm is somewhat dependent on the initial parameter settings it is useful to repeat the 'minimise the total travel distance' and 'minimise the longest tour' benchmarking experiments with a range of GA parameters, the results of which show the sensitivity of the genetic algorithm and TCX operator to initial parameter configurations.

We fix the number of salesman to $m = 10$ for the three test problems (A) MTSP-51, (B) MTSP-100 and (C) MTSP-150, while varying several GA parameters which typically impact the solution quality and performance of the algorithm, namely GA population ($P = 50, 200, 1000$), crossover probability ($Pc = 0.7,1.0$), mutation probability ($Pm = 0.02,0.001$). The default GA parameter settings are $P = 100$, $Pc = 0.85$ and $Pm = 0.01$.

From Table 11, the variation in GA population size shows an expected behaviour of the genetic algorithm. In general, a smaller population size ($P = 50$) results in lower solution quality and faster execution time while an increase ($P = 200, 1000$) results in higher or improved solution qualities and slower execution times for each of the three test problems. The variation of crossover probability also resulted in expected behaviour for the GA for all three test problems. A lower crossover probability ($Pc = 0.7$) reduces the exploration capability of the GA and results in a lower solution quality, while a high crossover probability ($Pc = 1.0$) can destroy good candidate solutions with too much exploration and not enough exploitation of the GA. The variation of mutation probability is more subtle with the results not being statistically significantly different from the default GA settings for all three test problems. The results suggest that a higher mutation probability ($Pm = 0.02$), which increases the amount of disruption to each gene in the chromosome, provides a slightly lower solution quality. However, a lower mutation probability ($Pm = 0.001$) produces a higher solution quality since the level of gene disruption is reduced by a factor of 10 on both parts of the chromosome.

From Table 12, the variation in GA population size shows an expected behaviour of the genetic algorithm. In general, a smaller population size ($P = 50$) results in lower solution quality and faster execution time while an increase ($P = 200, 1000$) results in higher or improved solution qualities and slower execution times for the more difficult test problems *B* and *C*. However, the solution quality for problem *A* does not vary greatly because the GA is able to find a near-optimal solution for each of the different population sizes. In general, a lower crossover probability ($Pc = 0.7$) reduces the exploration capability of the GA, but results in improved solution quality by maintaining good solutions in the population, while a high crossover probability ($Pc = 1.0$) can destroy good candidate solutions with too much exploration and not enough exploitation of the GA. In general, the results suggest that a slightly higher mutation probability ($Pm = 0.02$, default $Pm = 0.01$), produces a higher solution quality. However, a lower mutation probability ($Pm = 0.001$) produces a lower solution quality since the level of gene disruption is reduced by a factor of 10 on both parts of the chromosome.

Depending on the type of problem being solved, whether the total travel distance or the longest tour is minimised, the level of exploration versus exploitation is controlled by key genetic algorithm parameters which produce different solution qualities through their variation. The GA is able to robustly find near-optimal solutions that are typical of the GA parameter variations presented in this section.

7.4. Optimality gap

The TCX operator and GA presented in this paper are derived from the class of meta-heuristic algorithms used to find ***near-optimal*** solutions to the symmetric MTSP problem. However, it is interesting to compare the proposed approach to optimal solutions for a subset of small-sized MTSP problems, which include (MTSP-11a):11 cities derived from distances51, (MTSP-11b): all 11 cities from sp11_dist, (MTSP-12a): 12 cities derived from distances51, (MTSP-12b): all 12 cities from uk12_dist and (MTSP-16): contains 16 cities derived from 'distances51'. Each of these small-sized MTSP data sets is described using a symmetric distance (cost) matrix. More specifically, the distances51 data set was obtained from Professor Arthur E. Carter (Carter and Ragsdale, 2006), while the other data sets of sp11_dist and uk12_dist were obtained from http://people.sc.fsu.edu/~jburkardt/datasets/cities/cities.html. In addition, GA parameters are default values that are specified in Table 4, the number of generations is 50,000 and 30 independent trials were used for the TCX/GA experiments conducted in this section of the paper.

Computation of the optimal values for each problem was performed using the brute-force of an exhaustive search, which had CPU times several orders of magnitude greater than the proposed TCX/GA algorithm. Furthermore, we acknowledge that other exact solution methods such as that presented in Gavish and Srikanth (1986), which uses a branch-and-bound algorithm, seemed to provide exact solutions to several medium-sized MTSP problems that are described using either symmetric/asymmetric and Euclidean/non-Euclidean datasets.

The results presented in Tables 13 and 14, indicate that the genetic algorithm with the proposed TCX operator is able to find optimal and near-optimal solutions to a set of different MTSP problems.

7.5. Examination of TCX Operator with an additional dataset

Since the proposed TCX operator and genetic algorithm presented in this paper utilise the two-part chromosome structure presented in Carter and Ragsdale (2006), we focused our benchmarking effort of the different crossover operators onto the Carter data sets (MTSP-51, MTSP-100 and MTSP-150). This allowed for direct statistical comparison between Carter's results and those presented in this paper. However, it was interesting to repeat the experiments

Please cite this article in press as: Yuan, S., et al. A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2013.01.043

of the four crossover operators using a completely different symmetric MTSP distance (cost) matrix. As such, we used the sgb128 data set, which were obtained from http://people.sc.fsu.edu/~jburkardt/datasets/cities/cities.html and considers 128 cities in the continental US and lower Canada. The distance matrix considers the actual road distances between the particular cities.

The results in Table 15 for minimising the total travel distance suggest that the genetic algorithm with TCX operator achieve statistically significantly better solutions quality for all benchmarking tests when compared with the three alternative crossover techniques (ORX + A, CYX + A and PMX + A). Furthermore, in the benchmarking experiments minimising the longest tour, the GA and TCX operator achieve significantly better solution qualities for all experiments compared to the alternative methods as illustrated in Table 16. These results suggest that the proposed TCX operator robustly achieves good results in different benchmarking experiments.

8. Conclusion

This paper proposed a new crossover operator for solving the MTSP using GAs. We employed the existing two-part chromosome encoding technique, which has previously been shown to minimise the size of the search space, through the reduction of redundant candidate solutions. We found that the existing crossover operator for two-part chromosome encoding does limit the GAs' search ability and then showed that the proposed TCX can effectively enhance the search ability of the GA by generating more possible solutions.

We also evaluated and compared the proposed TCX operator with three existing crossover operators, namely ORX + A, CYX + A and PMX + A. The benchmarking was performed using two different MTSP objective functions, namely the objective of minimising total travel distance and the objective of minimising the makespan. The experimental results show that the TCX operator enables the GA to produce higher solution quality than the three existing crossover operators.

Our future work will focus on two main areas. Firstly, we would like to develop a parallel version of the GA and incorporate the TCX operator. This would greatly reduce the computation time and increase the solution quality for the same stopping criterion currently employed. Secondly, since many practical problems can be cast into the MTSP, we will incorporate the TCX operator into our existing GA for solving a large MTSP problem in scheduling at automated seaport terminals (Yuan et al., 2011).

Acknowledgments

We are grateful to Prof. A.E. Carter for providing the experimental dataset and advice on the MTSP. Without his help, this work would not have been possible. In addition, this work is supported by ARC Linkage Grant (LP0,882,745), Patrick Stevedores Holdings (PTS) and the University of Technology, Sydney, Australia (UTS). The authors acknowledge the valuable support provided by Professor Gamini Dissanayake (UTS), Dr. Binghuang Cai (UTS), Dr. Haye Lau (PTS) and Mr. Daniel Pagac (PTS).

References

Bektas, T., 2006. The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega 34, 209–219.

Bellmore, M., Hong, S., 1974. Transformation of multisalesmen problem to the standard traveling salesman problem. Journal of the Association Computer Machinery 21, 500–504.

Bo, Z.W., Hua, L.Z., Yu, Z.G., 2006. Optimization of process route by genetic algorithms. Robotics and Computer-Integrated Manufacturing 22, 180–188.

Carter, A.E., 2003. Design and Application of Genetic Algorithms for the Multiple Traveling Salesperson Assignment Problem. Unpublished PhD Research, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Carter, A.E., Ragsdale, C.T., 2006. A new approach to solving the multiple traveling salesperson problem using genetic algorithms. European Journal of Operational Research 175, 246–257.

Chatterjee, S., Carrera, C., Lynch, L.A., 1996. Genetic algorithms and traveling salesman problems. European Journal of Operational Research 93, 490–510.

Christofides, N., Mingozzi, A., Toth, P., 1981. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. Mathematical Programming 20, 255–282.

Cus, F., Balic, J., 2003. Optimization of cutting process by GA approach. Robotics and Computer-Integrated Manufacturing 19, 113–121.

Davis, L., 1985. Applying adaptive algorithms to epistatic domains. In: The International Joint Conference on Artificial Intelligence. IEEE Computer Society Press., Los Angeles, pp. 162–164.

DeJong, K., 1975. An Analysis of the Behaviour of a Class of Genetic Adaptive Systems. Unpublished PhD Thesis. University of Michigan, USA.

GAlib, 2007. A C++ Library of Genetic Algorithm Components, vol. 2011. <http://lancet.mit.edu/ga/>.

Gavish, B., Srikanth, K., 1986. An optimal solution method for large-scale multiple traveling salesmen problems. Operations Research 34, 698–717.

Gen, M., Cheng, R., 1997. Genetic Algorithms and Engineering Design. Wiley, New York.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley Longman, Boston, MA.

Goldberg, D.E., Lingle, J.R., 1985. Alleles, loci and the TSP. In The First International Conference on Genetic Algorithms and Their Applications, New Jersey, pp. 154–159.

Gorenstein, S., 1970. Printing press scheduling for multi-edition periodicals. Management Science 16, B373–383.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, Michigan.

Holland, J.H., 1992. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology. Control and Artificial Intelligence. MIT Press, Cambridge, MA.

Kellegöz, T., Toklu, B., Wilson, J., 2008. Comparing efficiencies of genetic crossover operators for one machine total weighted tardiness problem. Applied Mathematics and Computation 199, 590–598.

Kim, K.H., Park, Y.M., 2004. A crane scheduling method for port container terminals. European Journal of Operational Research 156, 752–768.

Kulkarni, A.J., Tai, K., 2010. Probability collectives: a multi-agent approach for solving combinatorial optimization problems. Applied Soft Computing 10, 759–771.

Liu, W., Li, S., Zhao, F., Zheng, A., 2009. An ant colony optimization algorithm for the multiple traveling salesmen problem. In: IEEE Conference on Industrial Electronics and Applications, pp. 1533–1537.

Malmborg, C., 1996. A genetic algorithm for service level based vehicle scheduling. European Journal of Operational Research 93, 121–134.

Oliver, I.M., Smith, D.J., Holland, J.R.C., 1987. A study of permutation crossover operators on the TSP. In The Second International Conference on Genetic Algorithms, New Jersey, pp. 224–230.

Park, Y.B., 2001. A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines. International Journal of Productions Economics 73, 175–188.

Reinelt, G., 2001. TSPLIB. <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>.

Ryan, J.L., Bailey, T.G., Moore, J.T., Carlton, W.B., 1998. Reactive tabu search in unmanned aerial reconnaissance simulations. In: The 30th Conference on Winter Simulation. IEEE Computer Society Press, Los Alamitos, CA, pp. 873–879.

Singh, A., Baghel, A.S., 2009. A new grouping genetic algorithm approach to the multiple traveling salesperson problem. Soft Computing 13, 95–101.

Smith, G.C., Smith, S.S.F., 2002. An enhanced genetic algorithm for automated assembly planning. Robotics and Computer-Integrated Manufacturing 18, 355–364.

Spears, W.M., DeJong, K.A., 1991. An analysis of multi-point crossover. In: Rawlins, J.E. (Ed.), Foundations of Genetic Algorithms. Morgan Kaufman, San Mateo, CA, pp. 301–315.

Svestka, J.A., Huckfeldt, V.E., 1973. Computational experience with an *m*-salesman traveling salesman algorithm. Management Science 17, 790–799.

Tang, L., Liu, J., Rong, A., Yang, Z., 2000. A multiple traveling salesman problem model for hot rolling scheduling in Shangai Baoshan Iron & Steel Complex. European Journal of Operational Research 124, 276–282.

Walpole, R.E., Myers, R.H., Myers, S.L., Ye, K., 2006. Probability & Statistics for Engineers & Scientists, eighth ed. Prentice Hall, Upper Saddle River, NJ.

Wang, X.B., Regan, A.C., 2002. Local truckload pickup and delivery with hard time window constraints. Transportation Research Part B 36, 97–112.

Xu, Z., Xu, L., Rodrigues, B., 2011. An analysis of the extended Christofides heuristic for the k-depot TSP. Operations Research Letters 39, 218–223.

Yuan, S., Skinner, B.T., Huang, S.D., Liu, D.K., Dissanayake, G., Lau, H., Pagac, D., 2011. A job grouping approach for planning container transfers at automated seaport container terminals. Advanced Engineering Informatics 25, 413–426.

Please cite this article in press as: Yuan, S., et al. A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. European Journal of Operational Research (2013), http://dx.doi.org/10.1016/j.ejor.2013.01.043