

An Immune System Based Genetic Algorithm Using Permutation-Based Dualism for Dynamic Traveling Salesman Problems

Lili Liu¹, Dingwei Wang¹, and Shengxiang Yang²

¹ School of Information Science and Engineering, Northeastern University
Shenyang 110004, P. R. China

liulili1202@gmail.com, dwwang@mail.neu.edu.cn

² Department of Computer Science, University of Leicester
University Road, Leicester LE1 7RH, United Kingdom
s.yang@mcs.le.ac.uk

Abstract. In recent years, optimization in dynamic environments has attracted a growing interest from the genetic algorithm community due to the importance and practicability in real world applications. This paper proposes a new genetic algorithm, based on the inspiration from biological immune systems, to address dynamic traveling salesman problems. Within the proposed algorithm, a permutation-based dualism is introduced in the course of clone process to promote the population diversity. In addition, a memory-based vaccination scheme is presented to further improve its tracking ability in dynamic environments. The experimental results show that the proposed diversification and memory enhancement methods can greatly improve the adaptability of genetic algorithms for dynamic traveling salesman problems.

1 Introduction

Many real-world optimization problems are time-varying, i.e., the fitness function, design variables and the environmental conditions may change over time due to the arrivals of new tasks, or circumstance fluctuations, and so on. It is interesting to investigate dynamic instances when changes are not so severe that the new optimal solutions might be related to the old ones. As one of the most typical NP-hard combinatorial optimization problems, the traveling salesman problem in dynamic environments (DTSP), which is caused by deletion of some cities, the change of traffic costs between cities, or the addition of cities over time, has a practical application in production and traffic systems [6,7,8] and requires optimization algorithms to be able to track the optima in new situations.

Genetic algorithms (GAs) have been applied to address dynamic optimization problems (DOPs) in recent years with several promising results [12,15,21]. Intrinsicly, GAs are motivated by the natural evolution, which is always subject to a changing environment. However, classic GAs should be modified to deal with the convergence problem for solving DOPs. The tendency of the standard

algorithm making all individuals eventually converge onto the promising individual of the current population may reduce the population diversity [1,2], which leads to a low exploration ability in dynamic environments. Therefore, several approaches have been developed into GAs to address DOPs, including random or guided immigrants [4,19] and hypermutation schemes [2].

Diversity and memory are two chief factors for enhancing the adaptability of organisms in the ever-changing environment in nature [20]. For example, the immune system of each animal body can generate antibodies to prevent invasions of detrimental pathogens [9]. As a major type of lymphocytes in the adaptive system, B-cells can be activated and cloned. The cloning process can be regarded as a special form of mutation, namely *somatic hypermutation*. In this way, the system holds various antibodies to neutralize antigens. Meanwhile, some B-cells can store the current information as memory in order to immediately respond to the same or similar pathogens. Hence, the diverse antibodies and the stored information help the immune system to detect and protect organisms against potentially harmful pathogens in different situations.

Following the above discussions, GAs with effective diversification and memory strategies inspired by the immune system, which can improve the exploration capacity and fully utilize the memorized information, may have a good potential in dynamic environments. Simões and Costa [18] proposed an immune system based GA, called *ISGA* for dynamic environments in the binary search space. It works by performing a transformation operation, which can be considered as the somatic hypermutation of B-cells, together with the memory scheme to allow the population rapidly respond to the changing landscapes. Yang [20] further investigated a variant of *ISGA* that adopts some new techniques, which greatly improve *ISGA*'s adaptivity in tracking the changing optima.

This paper extends the idea of the immune system based GAs to solve DTSPs. A permutation-based dualism scheme, which has proven to be a useful diversification technique for DOPs in the natural number-encoded space [15], is introduced in the clone process of the immune system based GA for DTSPs. The resultant GA is denoted *PISGA*. Moreover, a memory-based vaccination strategy is proposed with the expectation of evolving individuals by taking advantage of information stored in the past evolutionary process, which may lead to a better tracking capacity in dynamic environments. The effect of some key techniques on the performance of *PISGA* for DTSPs are experimentally investigated. Furthermore, the performance comparison with other GA variants validates the efficiency of *PISGA* for DTSPs.

2 Relevant Work on the Dynamic TSP

Dynamic TSP was firstly introduced by Psaraftis in [16]. In recent years, some variants of classic intelligent optimization methods have been developed for a promising behavior on DTSPs. Guntsch and Middendorf [8] introduced a population-based ant colony optimization algorithm to solve DTSPs. They applied and compared three strategies for modifying the pheromone information

instead of the restart strategy in reaction to the change of problem instances. Eyckelhof and Snoek [3] created DTSP instances by introducing “traffic jam” via increasing edge weights on the current best paths, and suggested that it is crucial to utilize the previous information for enhancing the search ability of ant systems in dynamic environments.

As a class of stochastic optimization technique that mimics the selection and modification principles in natural evolution, evolutionary algorithms (EAs) have attracted an increasing concern to address DTSPs. When addressing DTSPs, an efficient EA optimizer should show continuous adaption to track the changing optimal routes. Huang [10] carried out a thorough comparison between DTSP and some ordering problems to illustrate the universality and availability of this problem in real-world applications. The experimental results suggest that EAs might be a good candidate for DTSPs. Zhou [22] presented three types of operators based on some prior knowledge of the current environment to enhance the search ability of EAs for DTSPs. Li et al [13] further improved the inver-over operator based on a gene pool, which stores a set of most promising gene segments by applying some heuristic rules. These approaches indicate again that adopting effective diversity maintaining or enhancing schemes and memory-based methods may provide EAs with a competitive performance to solve DTSPs.

3 The Proposed Approach: PISGA

In this paper, a permutation-based dualism scheme and some mechanisms of the immune system are integrated into the standard GA for DTSPs. The proposed PISGA has the following three characteristics:

1. It shares the basic framework of the ISGA proposed in [20];
2. The diversification scheme utilizes the permutation-based dualism to simulate the somatic hypermutation when performing the clone operation;
3. The memory-based scheme employs a vaccination method by extracting and conveying valuable information according to the stored knowledge.

The pseudo-code of PISGA is outlined in Fig. 1. In the following subsections, the key operators and the framework of PISGA are described in details.

3.1 The Permutation-Based Dual Operator

In nature, some organisms can utilize pieces of DNA or gene segments from their complements to fully integrate diverse functions in response to different environments. Based on the idea that one organism or individual may have several complements or dual entities based on different attributes, a permutation-based dualism was firstly proposed to GAs in [15]. The principle of the dualism scheme is illustrated in Fig. 2. Similar to diploid mechanisms in GA, through the dual mapping based on different attributes, each individual has several dual entities regarding different attributes. This may promote the population diversity to a reasonable level.

```

t := 0
Initialize population P(0) and M(0) randomly
repeat
  Evaluate population P(t) and M(t)
  Replace the worst individual in P(t) by the elite from P(t - 1)
  if an environmental change is detected then
    Retrieve the best memory point B_M(t)
    Perform vaccination operations for r_v * n worst individuals by Eq. (4)
    t_M := t + rand(5, 10)
  end if
  if t = t_M then
    Replace the worst individual in M(t) with the best member in P(t)
  end if
  P'(t) := selectForReproduction(P(t))
  Performing clone operations for individuals in P'(t) using permutation-
    based dualism according to Eq. (3)
  t := t + 1
until the termination condition is met
    
```

Fig. 1. Pseudo-code of the PISGA, where t_M is the time point to update the memory

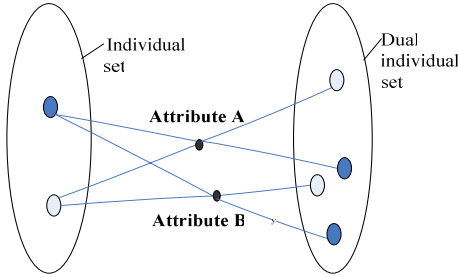


Fig. 2. Dual mapping based on different attributes

The concept of *permutation* is generally employed to investigate a particular family of groups, called *permutation group*, in the abstract algebra theory [5]. Since it can be dissolved into different cycles to represent collocation transformations, it is practical to introduce the permutation with some characteristics for decomposing the attributes. GAs with the permutation-based dual operator have been shown to outperform standard GA for solving some combinatorial optimization problems in dynamic environments [14,15].

Similar to the ISGA [20], PISGA starts from a population of B-cells. At each generation, individuals are selected to form the mating pool. Individuals in the mating pool will be randomly partitioned into two sub-populations, in which individuals are operated by the operator C_{dual} with respect to two attributes, namely discrete and inverse attribute respectively. They are described as follows:

1* *Discrete attribute*: when a m -bit chromosome S is transformed to S' by a permutation denoted as f_{d1} , not only all of their alleles are different, but also the neighborhood of each gene in S is shifted completely in S' . Permutation f_{d1} for discrete attribute is adopted as follows:

$$f_{d1} = \begin{cases} (1, 3, \dots, 2i - 1, \dots, 2(m - i), \dots, 4, 2), & \text{if } m \text{ is even,} \\ (1, 3, \dots, 2i - 1, \dots, 2(m - i + 1), \dots, 4, 2), & \text{if } m \text{ is odd,} \end{cases} \quad (1)$$

where i denotes the i -th bit of the chromosome.

2* *Inverse attribute*: when a chromosome S is transformed to S' by a permutation denoted as f_{d2} , the order of genes in S is inverted in S' . The permutation f_{d2} for the inverse attribute is designed as follows:

$$f_{d2} = \omega'_1 \omega'_2 \dots \omega'_i \dots \omega'_{[m/2]} \quad (2)$$

where ω'_i is the transposition $(i, m - i + 1)$ and “[]” denotes rounding the inner value down to an integer.

Given the above definitions, a dual operator can be defined as follows:

$$C_{dual} = (\alpha_1, d'_1)(\alpha_2, d'_2) \dots (\alpha_s, d'_s) \quad (3)$$

where $\alpha_i \in [0, 1]$ is the dual factor and the transposition d'_i , called the *dual unit*, takes effect in the dual operation with a probability α_i . The selected individual is mapped onto a new individual by C_{dual} .

We adopt such an operator for two reasons. First, one permutation can be dissolved into several cycles. Each individual can be transformed by these cycles with the corresponding dual factors and a permutation can create k ($k \in [1, \sum_{i=1}^s C_s^i]$) dual individuals. Hence, the diversity degree might be greatly improved even if the population has been convergent. Second, with the dual factor, some adjacency relationships of cities may be preserved into the next generation, which is expected to provide continuous tracking capacity over time.

3.2 The Memory-Based Vaccination

Jiao and Wang [11] introduced the vaccination which is abstracted from the prior knowledge of a problem, with the aim of accelerating the search speed for seeking the fixed optimum in a static context.

In this study, a novel vaccination extraction and injection technique based on the stored information is proposed in PISGA, which is also inspired by the permutation group theory. PISGA uses a memory of size N_m with a stochastic updating period (i.e., if an updating occurs at time t , the next time to update the memory is $t_M := t + rand(5, 10)$), and the vaccination operation is evoked when an environment change is detected. The detection method is similar to that used in Yang’s ISGA [20]. That is, the stored individuals are re-evaluated for each generation, and a change is considered to occur if the fitness value of any individual does not match its memorized best value at the last generation. In PISGA, the stored individual which mostly matches the current environment

is retrieved and utilized to perform vaccination operations for the $r_v * n$ worst members, where n is the population size and r_v is the vaccination ratio. The vaccination operation is described below.

If we denote a chromosome as $S = (c_1, c_2, \dots, c_m)$ and the best memory solution as $S_m = (a_1, a_2, \dots, a_m)$, there exists a permutation f_v :

$$f_v : \boxed{c_1} \boxed{c_2} \dots \boxed{c_m} \rightarrow \boxed{a_1} \boxed{a_2} \dots \boxed{a_m}$$

where f_v is equal to a product of disjointed cycles. Hence, the vaccine with respect to the memory point S_m can be defined as follows:

$$C_{vaccine} = (\beta_1, v'_1)(\beta_2, v'_2) \dots (\beta_k, v'_k) \quad (4)$$

where $\beta_i \in [0, 1]$ is the vaccine-extraction probability and the cycle v'_i is the vaccine unit that takes effect in the vaccination operation with the probability β_i . The selected individual is transformed to a new individual by $C_{vaccine}$.

The reasons for introducing this scheme lies in two factors. First, it is difficult to obtain the knowledge of a problem in advance, especially for the real-world applications in dynamic contexts. The best memory point which may reflect the requirements of the current environment in some extent may be a good candidate for extracting vaccine for a better optima tracking capacity. Second, the injection operation considers part of “promising” cycles instead of all of them. It is expected to help individuals to take in valuable information and prevent converging to the best memory point.

4 Experimental Study

4.1 Experimental Design

In order to test the proposed PISGA, experiments were carried out on DTSPs to compare the performance of PISGA with other two variants of GAs for dynamic environments. The first GA, denoted *R-SGA*, re-initializes all individuals when a change is detected, which is a straightforward method to respond to dynamic environments. Another peer GA is the permutation-based dual GA (PBDGA) [15], which has proven to be competitive in some dynamic environments.

In this study, we adopted the method proposed in [8] to create DTSP instances based on the data of kroA100 [17], which is created by changing the costs between some cities. That is, randomly removing half of the cities from the standard TSP data set to create a *spare pool*. For every τ intervals, a set of c cities are randomly replaced by the locations in the spare pool. For the purpose of testing the adaption of algorithms under different dynamics, the environmental dynamics parameters were set to $c \in \{1, 10, 15\}$ and $\tau \in \{50, 100, 200\}$, where c determines the severity of changes and τ specifies the speed of changes. This gives 9 dynamic scenarios as listed in Table 1. For each experiment on one scenario, 10 environmental changes were allowed for each algorithm on a DTSP.

For all GAs, parameters were set as follows: the uniform cycle crossover with a probability $p_c = 0.6$, removal mutation with the mutation probability $p_m = 0.1$,

Table 1. The t -test results of comparing algorithms on dynamic problems

(τ, c)	(50,1)	(50,10)	(50,15)	(100,1)	(100,10)	(100,15)	(200,1)	(200,10)	(200,15)
Scenario	1	2	3	4	5	6	7	8	9
PISGA – R-SGA	+	+	+	+	+	+	+	+	+
PISGA – PBDGA	~	~	~	+	+	+	+	+	+

fitness proportionate selection with the roulette wheel and elitism strategies [10]. The population size is set to 100, including 10 individuals stored in the memory. For PISGA and PBDGA, all the dual probabilities and vaccination probabilities were set to 0.5 and the vaccination ratio r_v was set to 0.1.

The overall performance of algorithms is defined as follows [19]:

$$\overline{F_{BG}} = \frac{1}{NG} \sum_{i=1}^{NG} (F_{BGi}) \quad (5)$$

where $\overline{F_{BG}}$ is the mean best-of-generation fitness averaged across the whole evolutionary process, F_{BGi} is the best-of-generation fitness of generation i , NG is the total number of generations.

4.2 Experimental Results and Analysis

The experimental results for comparisons are provided in Fig. 3 on different dynamic instances. The corresponding statistical test results of comparing algorithms by one-tailed t -test with 98 degrees of freedom at a 0.05 level of significance are given in Table 1, where the result regarding Alg. 1 – Alg. 2 is shown as “+”, “-”, or “~” when Alg. 1 is significantly better than, significantly worse than, or statistically equivalent to Alg. 2 respectively.

From Fig. 3 and Table 1, it can be seen that PISGA outperforms R-SGA and PBDGA in most test cases. This result validates our expectation of the permutation-based dual mechanism and the memory-based vaccination approach in the immune-based GA for investigated DTSPs. Comparing PISGA and PBDGA, an interesting observation is that PISGA dominates on investigated DTSPs especially with low change frequencies (i.e., $\tau = 100$ and 200). The reason may lie in that sufficient search iterations which could produce a better memory point in the past is beneficial for tracking the optimum in a new environment. R-SGA performs the worst in all cases. This demonstrates that the combination of the dual mechanism and other approaches that enhance the exploitation ability in promising regions, such as the learning operator in PBDGA and the vaccination operator in PISGA, may lead to a better tracking behavior for GAs.

One technique introduced in PISGA is the permutation-based dual scheme. In order to investigate the effect of this operator on the performance of PISGA, experiments were implemented on PISGA with the dual probability α setting to 0, 1 and 0.5, which implies performing clone without dual operator, completely transforming the gene segments of individuals in the mating pool with respect to the two attributes, and partially transforming the gene segments with the probability of 0.5 respectively. Experimental results on the dynamic behavior of

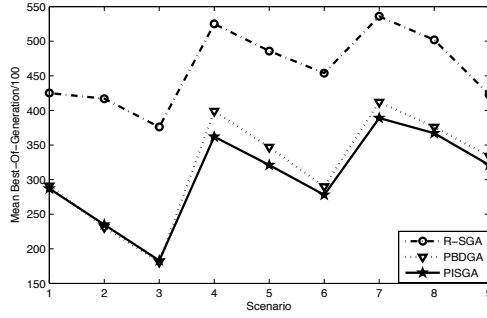


Fig. 3. Mean best-of-generation fitness for different DTSPs

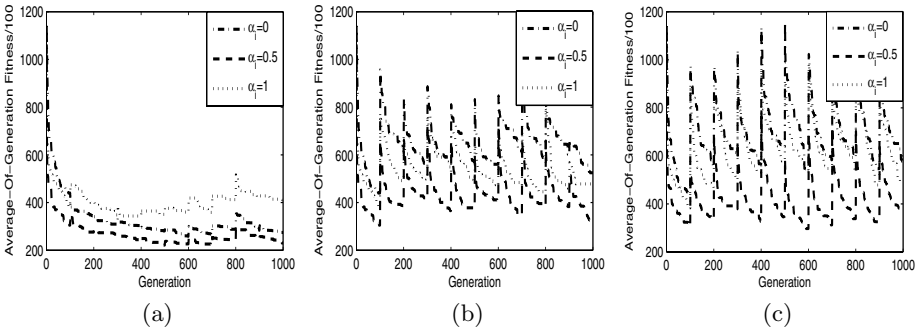


Fig. 4. Dynamic performance of PISGA with different α_i on investigated DTSPs with $\tau = 100$ and different severities of environmental changes: (a) $c = 1$, (b) $c = 10$, and (c) $c = 15$

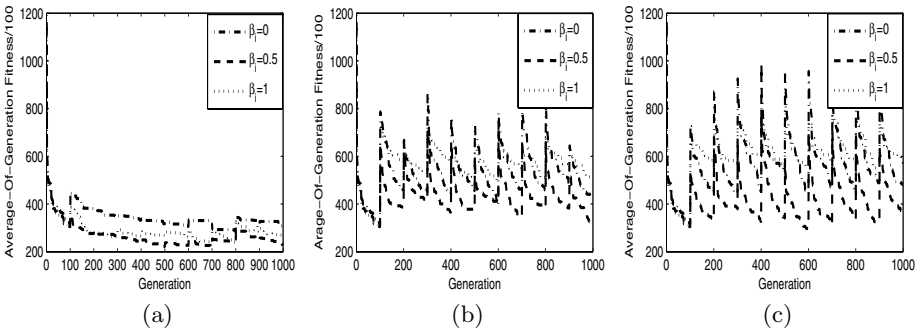


Fig. 5. Dynamic performance of PISGA with different β_i on investigated DTSPs with $\tau = 100$ and different severities of environmental changes: (a) $c = 1$, (b) $c = 10$, and (c) $c = 15$

PISGAs on investigated DTSPs with $\tau = 100$ are given in Fig. 4. It can be seen that the value of 0.5 always perform the best, indicating an advantage of the adopted dual mechanism over the other two settings for investigated DTSPs. PISGA with $\alpha = 0$ tends to strongly suffer from being trapped to local optima due to the diversity loss. The convergence curve of PISGA with $\alpha = 1$ shows that a reasonable diversity degree (i.e., not the larger, the better) is more helpful for a better performance in dynamic environments.

Furthermore, to examine the efficiency of the vaccination scheme on the behavior of PISGA, experiments were also carried out with the the value of β set to 0, 1, and 0.5 on the DTSPs. The results, presented in Fig. 5, show that the proposed memory-based approach is necessary and that a large β (i.e., $\beta = 1$) may weaken the search efficiency due to the diversity deficiency.

5 Conclusions and Future Work

This paper proposes an immune system based genetic algorithm to address dynamic traveling salesman problems. A permutation-based dualism scheme is introduced and integrated into the immune operation in order to enhance the exploration capacity of the population to watch over new optima. Furthermore, a memory-based vaccination strategy is presented to drive individuals to exploit promising regions based on the valuable information extracted from the memory. Experimental results show the efficiency of the proposed techniques for the immune-based GA for DTSPs.

There are several issues deserving the future work. A further study on the sensitivity of parameters on the performance of PISGA and a comprehensive comparison with other approaches for DTSPs are under investigation. It would be valuable to introduce other mechanisms in immune systems for GAs in dynamic environments. Another interesting topic is to extend our scope to more complex dynamic problems, such as dynamic vehicle routing problems.

Acknowledgments

The work by Lili Liu and Dingwei Wang was supported by the Key Program of National Natural Science Foundation (NNSF) of China under Grant No. 70431003 and Grant No. 70671020, the Science Fund for Creative Research Group of NNSF of China under Grant No. 60521003 and the National Science and Technology Support Plan of China under Grant No. 2006BAH02A09. The work by Shengxiang Yang was supported by the Engineering and Physical Sciences Research Council (EPSRC) of UK under Grant No. EP/E060722/1.

References

1. Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Proc. of the 1999 Congr. on Evol. Comput., vol. 3, pp. 1875–1882 (1999)
2. Cobb, H.G., Grefenstette, J.J.: Genetic algorithms for tracking changing environments. In: Proc. of the 5th Int. Conf. on Genetic Algorithms, pp. 523–530 (1993)

3. Eyckelhof, C.J., Snoek, M.: In: ANTS 2002: Proc. of the 3rd Int. Workshop on Ant Algorithms, pp. 88–99 (2002)
4. Grefenstette, J.J.: Genetic algorithms for changing environments. In: Proc. of the 2nd Int. Conf. on Parallel Problem Solving from Nature, pp. 137–144 (1992)
5. Gilbert, W.J.: Modern Algebra with Application. John Wiley and Sons, Chichester (1976)
6. Guntsch, M., Middendorf, M., Schmeck, H.: An ant colony optimization approach to dynamic TSP. In: Proc. of the 2001 Genetic and Evol. Comput. Conf., pp. 860–867 (2000)
7. Guntsch, M., Middendorf, M.: Pheromone modification strategies for ant algorithms applied to dynamic TSP. In: Boers, E.J.W., et al. (eds.) EvoWorkshops 2001. LNCS, vol. 2037, pp. 213–222. Springer, Heidelberg (2001)
8. Guntsch, M., Middendorf, M.: A population based approach for ACO. In: Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G.R. (eds.) EvoWorkshops 2002. LNCS, vol. 2279, pp. 72–81. Springer, Heidelberg (2002)
9. Hartl, D.L., Jones, E.W.: Genetics: Principles and Analysis. Jones and Bartlett Publishers, Inc. (1998)
10. Huang, Z., Hu, X., Chen, S.: Dynamic traveling salesman problem based on evolutionary computation. In: Proc. of the 2001 IEEE Congr. on Evol. Comput., pp. 1283–1288 (2001)
11. Jiao, L., Wang, L.: A novel genetic algorithm based on immunity. IEEE Trans. on Systems, Man, and Cybern. Part A: Systems and Humans 30(5), 552–561 (2000)
12. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments - A survey. IEEE Trans. on Evol. Comput. 9(6), 303–317 (2005)
13. Li, C., Yang, M., Kang, L.: A new approach to solving dynamic traveling salesman problems. In: Wang, T.-D., Li, X.-D., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G.-L., Yao, X. (eds.) SEAL 2006. LNCS, vol. 4247, pp. 236–243. Springer, Heidelberg (2006)
14. Liu, L., Wang, D., Wang, H.: A new dual scheme for genetic algorithm in dynamic environments. In: Proc. of the 2008 Control and Decision Conf., pp. 135–138 (2008)
15. Liu, L., Wang, D., Ip, W.H.: A permutation-based dual genetic algorithm for dynamic optimization problems. In: Soft Computing (published online on July 18, 2008)
16. Psaraftis, H.N.: Dynamic vehicle routing problems. In: Golden, B.L., Assad, A.A. (eds.) Vehicle Routing: Methods and Studies, pp. 223–248. Elsevier, Amsterdam (1988)
17. Reinelt, G.: TSPLIB. University of Heidelberg (1996), <http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/>
18. Simões, A., Costa, E.: An immune system-based genetic algorithm to deal with dynamic environments: diversity and memory. In: Proc. of the 6th Int. Conf. on Neural Networks and Genetic Algs., pp. 168–174 (2003)
19. Yang, S.: Memory-based immigrants for genetic algorithms in dynamic environments. In: Proc. of the 2005 Genetic and Evol. Comput. Conf., pp. 1115–1122 (2005)
20. Yang, S.: A comparative study of immune system based genetic algorithms in dynamic environments. In: Proc. of the 2006 Genetic and Evol. Comput. Conf., pp. 1377–1384 (2006)
21. Yang, S., Yao, X.: Experimental study on population-based incremental learning algorithms for dynamic optimization problems. Soft Comput. 9(11), 815–834 (2005)
22. Zhou, A., Kang, L., Yan, Z.: Solving dynamic TSP with evolutionary approach in real time. In: Proc. of the 2003 IEEE Congr. on Evol. Comput., pp. 951–957 (2003)