Mixed strategy may outperform pure strategy: An initial study

He, Jun; Hou, Wei; Dong, Hongbin; He, Feidun

Published in:
2013 IEEE Congress on Evolutionary Computation (CEC)

DOI:
10.1109/CEC.2013.6557618

Publication date:
2013

Citation for published version (APA):
He, J., Hou, W., Dong, H., & He, F. (2013). Mixed strategy may outperform pure strategy: An initial study. In *2013 IEEE Congress on Evolutionary Computation (CEC)* (pp. 562-569). IEEE Press. https://doi.org/10.1109/CEC.2013.6557618

Mixed Strategy May Outperform Pure Strategy: An Initial Study

Jun He, Wei Hou, Hongbin Dong and Feidun He

Abstract

A pure strategy metaheuristics is one that applies the same search method at each iteration of the algorithm. A mixed strategy metaheuristics is one that selects a search method probabilistically from a set of strategies at each iteration. For example, a search strategy may be mutation + selection or crossover + selection. Thus classical genetic algorithms (using mutation, crossover and selection) belong to mixed strategy heuristics, while simplified (1+1) evolutionary algorithms (using mutation and selection) are pure strategy metaheuristics. The aim of this paper is to compare the performance between mixed strategy and pure strategy metaheuristics. The major results of the current paper are summarised as follows. (1) We construct two novel mixed strategy evolutionary algorithms for solving the 0-1 knapsack problem and show that the mixed strategy algorithms may find better solutions than pure strategy algorithms in up to 77.8% instances through experiments. (2) We establish a sufficient and necessary condition when the runtime of mixed strategy metaheuristics is smaller that that of pure strategy mixed strategy metaheuristics.

Index Terms

Hybrid Meta-heuristics, Performance Comparison, Theoretical analysis, Mixed Strategy, Pure Strategy

I. Introduction

In the last three decades, meta-heuristic algorithms have been widely applied in solving combinatorial optimisation problems. Meta-heuristics include, but are not restricted to, Ant Colony Optimization (ACO), Genetic Algorithms (GA), Iterated Local Search (ILS), Simulated Annealing (SA), and Tabu Search (TS) [1]–[3]. Different search strategies have been developed in these metaheuristics. Each search strategy has its own advantage. Therefore it is a natural idea to combine the advantages of several search strategies together. This leads to hybrid metaheuristics [4] such as hyper-heuristic [5] and memetic algorithm [6].

Mixed strategy heuristics [7] belong the family of hybrid metaheuristics. They are inspired from mixed strategies in the game theory [8]. A pure strategy metaheuristics is one that applies the same search method at each iteration of the algorithm. A mixed strategy metaheuristics is one that selects a search method probabilistically from a set of strategies at each iteration. For example, a search strategy may be mutation + selection or crossover + selection. Thus classical genetic algorithms (using mutation with probability 0.9 and crossover with probability 0.1) belong to mixed strategy heuristics. Simplified (1+1) evolutionary algorithms (using mutation) are pure strategy metaheuristics. Mixed strategy evolutionary programming, integrating several mutation operators, has been designed for numerical optimization [9]. Experimental results show that mixed strategy evolutionary programming outperforms pure strategy evolutionary programming with a single mutation operator [10].

The first goal of this paper is to conduct an empirical comparison of the performance between mixed strategy and pure strategy evolutionary algorithms (EAs for short) on the 0-1 knapsack problem. Novel mixed strategy EAs are proposed to solve the problem. In experiments, the performance of an EA is measured by the final fitness value after 500 generations averaged over 10 runs.

The second but more important goal is to provide a theoretical answer to the question: when do mixed strategy metaheuristics outperform pure strategy metaheuristics? In theoretical analysis, the performance of an algorithm is measured by the expected hitting time to find an optimal solution.

Despite the popularity of hybrid metaheuristics, there are few rigorous analyses of hybrid metaheuristics. One result is based on the asymptotic convergence rate as the performance measure [11]. It demonstrates that any mixed strategy (1+1) EA (consisting of several mutation operators) performs no worse than the worst pure strategy EA (using a single mutation operator). If mutation operators are mutually complementary, then it is possible to design a mixed strategy (1+1) EA better than the best pure strategy (1+1) EA. Another result is based on the runtime analysis of selection hyper-heuristics [12]. It shows that mixing different neighbourhood or move acceptance operators can be more efficient than using stand-alone individual operators in some cases. But the discussion is restricted to simple algorithms for the OneMax and GapPath functions.

There are two major differences between this paper and our previous work [11]. In this paper, the performance of an algorithm is theoretically measured by the expected hitting time. Nonetheless the asymptotic convergence rate is taken as the

This work is supported by the EPSRC under Grant EP/I009809/1, the National Natural Science Foundation of China under Grant(60973075) and Ministry of Industry and Information Technology (B0720110002).

Jun He is with Department of Computer Science, Aberystwyth University, Ceredigion, SY23 3DB, UK. Email: jqh@aber.ac.uk.
Feidun He is with School of Information Science and Technology, Southwest Jiaotong University, Chengdu, Sichuan, 610031, China
Wei Hou and Hongbin are Dong with College of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China

masure in [11]. Roughly speaking, the asymptotic convergence rate is how fast an iteration algorithm converge to the solution per iteration [13]. The algorithms discussed in the paper are population-based, but only (1+1) EAs are analysed in [11].

The rest of this paper is organized as follows. Section II gives experimental results that show mixed strategy may outperform pure strategy. Section III provides the sufficient and necessary condition when mixed strategy may outperform pure strategy in general. Section IV concludes the paper.

II. Evidence from Experiment: Mixed Strategy May Outperform Pure Strategy

This section conducts an empirical comparison of the performance between mixed strategy EAs and pure strategies EAs. A classical NP-hard problem, the 0-1 knapsack problem [14], is used in the empirical study.

A. Evolutionary Algorithms for the 0-1 Knapsack Problem

The 0-1 knapsack problem is described as follows:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^n v_i x_i, \\ \text{subject to} & \sum_{i=1}^n w_i x_i \le C, \end{array}$$

where $v_i > 0$ is the value of item i, $w_i > 0$ the weight of item i, and $C > 0$ the knapsack capacity and

$$x_i = \begin{cases} 1 & \text{if item } i \text{ is selected in the knapsack,} \\ 0 & \text{otherwise.} \end{cases}$$

A solution is represented by a vector (a binary string) $\vec{x} = (x_1, \cdots, x_n)$. If a solution $\vec{x}$ violates the constraint, then it is called infeasible. Otherwise it is called feasible.

There are several ways to handle the constrains in the knapsack problem [15]. The method of repairing infeasible solutions is used in the paper since it is more efficient than other methods [16]. Its idea is simple: if an infeasible solution is generated, then it will be repaired to a feasible solution. The repairing procedure is described as follows:

input $\vec{x}$;
if $\sum_{i=1}^n x_i w_i > C$ **then**
  $\vec{x}$ is infeasible;
  while ($\vec{x}$ is infeasible) **do**
    $i =$: **select** an item from the knapsack;
    set $x_i = 0$;
    if $\sum_{i=1}^n x_i w_i \le C$ **then**
      $\vec{x}$ is feasible;
    end if
  end while
end if
output $\vec{x}$.

There are different **select** methods in the repairing procedure. Two of them are described as follows.

1) *Random repair:* select an item from the knapsack at random and remove it from the knapsack.
2) *Greedy repairing:* sort all items according to the order of the ratio v_i/w_i, then select the item with the smallest ratio and remove it from the knapsack.

The fitness function is defined as

$$f(\vec{x}) = \sum_{i=1}^n x_i v_i, \text{ if } \vec{x} \text{ is feasible,}$$

Thanks to the repairing method, no need to define the fitness for infeasible solutions.

A pure strategy EA for solving the 0-1 knapsack problem is described as follows.

input a fitness function;
generation counter $t \leftarrow 0$;
initialize Φ_0;
an archive keeps the best solution in Φ_0;
while t is less than a threshold **do**
  $\Phi_{t+1/2} \leftarrow$ children mutated from Φ_t;
  if a child is an infeasible solution **then**
    then repair it into a feasible solution;
  end if
  $\Phi_{t+1} \leftarrow$ selected from $\Phi_t, \Phi_{t+1/2}$;
  update the archive if the best solution in Φ_{t+1} is better than it;

$t \leftarrow t+1$;

end while

output the maximum of the fitness function.

A mixed strategy EA for solving the 0-1 knapsack problem is almost the same as the above algorithm, except one place:

choose a mutation operator probabilistically;

$\Phi_{t+1/2} \leftarrow$ children mutated from Φ_t;

A detailed introduction of mutation operators is given in the next subsection. The selection operator is the same in all pure strategy and mixed strategy EAs. Therefore a pure strategy refers to a mutation operator. A mixed strategy then means a probability distribution of choosing mutation operators.

B. Pure Strategy and Mixed Strategy Evolutionary Algorithms

Four pure strategy EAs are constructed based on four different mutation operators. One operator is independent on the 0-1 knapsack problem and the others are problem-specific.

The first mutation operator is standard bitwise mutation. The related EA is denoted by PSb.

- *Bitwise Mutation:* Flip each bit x_i to $1 - x_i$ with probability $\frac{1}{n}$.

The second mutation operator is problem-specific. It is based on heuristic knowledge: an item with a bigger value is more likely to appear in the knapsack. The related EA is denoted by PSv.

- *Mutation based on values:* If a bit $x_i = 0$, then flip it to 1 with probability

$$\frac{v_i}{\sum_{j=1}^n v_j}. \tag{1}$$

If a bit $x_i = 1$, then flip it to 0 with probability

$$\frac{1/v_i}{\sum_{j=1}^n 1/v_j}. \tag{2}$$

The third mutation operator is based on heuristic knowledge too: an item with a smaller weight is more likely to appear in the knapsack. The corresponding EA is denoted by PSw.

- *Mutation based on weights:* If a bit $x_i = 0$, then flip it to 1 with probability

$$\frac{1/w_i}{\sum_{j=1}^n 1/w_j}. \tag{3}$$

If a bit $x_i = 1$, then flip it to 0 with probability

$$\frac{w_i}{\sum_{j=1}^n w_j}. \tag{4}$$

The fourth mutation operator is constructed from heuristics knowledge: first calculate the ratio between the value and weight for each item:

$$r_i = \frac{v_i}{w_i}. \tag{5}$$

Then an item with a bigger ratio is more likely to appear in the knapsack. The related EA is denoted by PSr.

- *Mutation based on the ratio between value and weight :* If a bit $x_i = 0$, then flip it to 1 with probability

$$\frac{r_i}{\sum_{j=1}^n r_j}. \tag{6}$$

If a bit $x_i = 1$, then flip it to 0 with probability

$$\frac{1/r_i}{\sum_{j=1}^n 1/r_j}. \tag{7}$$

Two types of mixed strategy are designed in the experiments. One is to set a fixed probability distribution of choosing mutation operators for all time. The algorithm is called *static*, denoted by MSs.

- *statically mixed strategy:* choose each mutation strategy subject to a fixed probability distribution. In the experiments, we set the probability distribution to be $(0.25, 0.25, 0.25, 0.25)$ for the four pure strategies.

The other is to dynamically adjust the probability distribution of choosing mutation operators. If a better solution is generated by applying a mutation operator this time, then the operator will be chosen with a higher probability next time. This kind of mixed strategy EAs is called *dynamic*, denoted by MSd.

- *dynamically mixed strategy:* The mixed strategy used in the experiments is the same as that in [9]. For each individual in population $I(t+1)$, its mixed strategy should be adjusted as follows:
  - If individual i comes from the offspring population $I'(t)$, or comes from the parent population $I(t)$ and its offspring appears in $I(t+1)$, the pure strategy used in the mutation is strategy h, then the new mixed strategy is given by:
$$\begin{cases} \rho_i^{(t+1)}(h) = \rho_i^{(t)}(h) + (1-\rho_i^{(t)}(h))\gamma \\ \rho_i^{(t+1)}(l) = \rho_i^{(t)}(l) - \rho_i^{(t)}(l)\gamma, & \forall l \neq h \end{cases} \tag{8}$$
    where the parameter $\gamma \in (0,1)$ is used to control the mixed strategy distribution $\vec{\rho}_i$. It is chosen to be a positive in $(0,1)$ to guarantee the normalization condition: $\sum_{k=1}^{4} \rho_i(k) = 1$ and $\rho_i(k) \geq 0$. In this paper, γ is set to $1/3$.
  - If individual i comes from the parent population $I(t)$ but its offspring does not appear in $I(t+1)$, and the pure strategy used in the mutation is strategy h, then we weaken the strategy h. The new probability distribution is given by:
$$\begin{cases} \rho_i^{(t+1)}(h) = \rho_i^{(t)}(h) - \rho_i^{(t)}(h)\gamma \\ \rho_i^{(t+1)}(l) = \rho_i^{(t)}(l) + \rho_i^{(t)}(l)\gamma, & \forall l \neq h \end{cases} \tag{9}$$

C. Experiments

Experiments are conducted on different types of instances of the 0-1 knapsack problem. According to the correlation between values and weights, the instances of the problem are classified into three types [14], [15]: given two positive parameters A and B,

1) *uncorrelated knapsack*: v_i and w_i uniformly random in $[1, A]$;
2) *weakly correlated knapsack*: w_i uniformly random in $[1, A]$; and v_i uniformly random in $[w_i - B, w_i + B]$ (if for some j, $v_i \leq 0$, then the random generation procedure should be repeated until $v_i > 0$);
3) *strongly correlated knapsack*: w_i uniformly random in $[1, A]$; and $v_i = w_i + B$;

In the experiments, A and r are set to be $A = \frac{n}{20}$ and $B = \frac{n}{20}$.

Based on the capacity, the instances of the knapsack problem are classified into two types [14], [15]:

1) *restrictive capacity knapsack:* the knapsack capacity is small, where $C = 2A$.
2) *average capacity knapsack:* the knapsack capacity is large, where $C = 0.5 \sum_{i=1}^{n} w_i$.

Hence we will compare two mixed strategy EAs and four pure strategy EAs on six different types of instances below:

1) uncorrelated and restrictive capacity knapsack,
2) weakly correlated and restrictive capacity knapsack,
3) strongly correlated and restrictive capacity knapsack,
4) uncorrelated correlated and average capacity knapsack,
5) weakly correlated and average capacity knapsack,
6) strongly correlated and restrictive average capacity knapsack.

Furthermore the experiments are split into two groups based on repairing methods: (1) greedy repair, (2) random repair.

The experiment setting is described as follows. For each type of the 0-1 knapsack problem, three instances with 100, 250 and 500 items are generated at random. The population size is set to 10. The maximum of generations is 500. The initial population is chosen at random. Tables I and II give the fitness values of the archive after 500 generations. It is averaged over 10 independent runs.

Following a simple calculation, we see that the dynamically mixed strategy EA, MSd, is the best in 77.8% instances and equally the best in 2.8% instances. Comparing the statically mixed strategy EA, MSs, with four pure strategies, MSs is better in 36.1% instances (marked in italic type in the tables).

Experimental results show mixed strategy EAs outperform pure strategy EAs in many instances, but not always. Naturally it raises the question: under what condition, a mixed strategy EA may outperform a pure strategy EA. This question is seldom answered rigorously before.

III. Support of Theory: Mixed Strategy May Outperform Pure Strategy

In this section, we conduct a theoretical comparison of the performance between mixed strategy metaheuristics and pure strategy metaheuristics.

TABLE I
GREEDY REPAIR: FITNESS VALUES AFTER 500 GENERATIONS AVERAGED OVER 10 RUNS

uncorrelated and restrictive capacity knapsacks						
n	MSs	MSd	PSb	PSv	PSw	PSr
100	*285*	**300**	279	281	283	277
250	*1609*	**1655**	1601	1539	1528	1513
500	5625	5703	5515	**5794**	5140	5504
weakly correlated and restrictive capacity knapsacks						
n	MSs	MSd	PSb	PSv	PSw	PSr
100	*342*	**353**	331	289	306	325
250	1651	**1695**	1583	1514	1668	1650
500	*5319*	**5545**	5161	4595	4810	4710
strongly correlated and restrictive capacity knapsacks						
n	MSs	MSd	PSb	PSv	PSw	PSr
100	671	**683**	678	662	655	665
250	4126	4261	4212	4170	**5023**	3980
500	15273	**15537**	14959	15226	15367	14179
uncorrelated and average capacity knapsacks						
n	MSs	MSd	PSb	PSv	PSw	PSr
100	295	**299**	293	292	288	295
250	1616	**1650**	1616	1619	1583	1609
500	5751	5958	5670	**5963**	5601	5663
weakly correlated and average capacity knapsacks						
n	MSs	MSd	PSb	PSv	PSw	PSr
100	*387*	**395**	355	344	362	349
250	1976	**2014**	2009	1924	1997	1956
500	*7284*	**7505**	6839	6966	7048	7049
strongly correlated and average capacity knapsacks						
n	MSs	MSd	PSb	PSv	PSw	PSr
100	716	**730**	718	721	714	716
250	*4372*	**4409**	4301	4220	4135	4218
500	15525	**15868**	15746	15819	15552	14828

A. Meta-heuristics and Markov Chains

Without losing generality, consider the problem of maximising a fitness function:

$$\text{maximize } f(x), \tag{10}$$

where x is a variable and its definition domain is a finite set.

The metaheuristics considered in the paper are formalised as Markov chains. Initially construct a population of solutions Φ_0; from Φ_0, then generate a new population of solutions Φ_1; from Φ_1, then generate a new population of solutions Φ_2, and so on. This procedure is repeated until a stopping condition is satisfied. A sequence of populations is then generated

$$\Phi_0 \to \Phi_1 \to \Phi_2 \to \cdots .$$

An archive is used for recording the best found solution so far. The archive is not involved in generating a new population. In this way, the best found solution is preserved for ever (called *elitist*). The metaheuristics algorithm with an archive is described below.

set counter t to 0;
initialize a population Φ_0;
an archive keeps the best solution in Φ_0;
while the archive is not an optimal solution **do**
    a new population Φ_{t+1} is generated from Φ_t;
    update the archive if the best solution in Φ_{t+1} is better than it;
    counter t is increased by 1;
end while

The procedure of generating Φ_{t+1} from Φ_t can be represented by transition probabilities among populations:

$$P(X,Y) := P(\Phi_{t+1} = Y \mid \Phi_t = X), \tag{11}$$

where populations Φ_t, Φ_{t+1} are variables and X, Y are their values (also called states). The transition probabilities $P(X,Y)$ form the transition matrix of a Markov chain, denoted by $\mathbf{P}$.

Definition 1: If a transition matrix $\mathbf{P}$ for generating new populations is independent of t, then it is called a *pure strategy*. A *mixed strategy* is a probability distribution of choosing pure strategies over a strategy pool.

TABLE II
RANDOM REPAIR: FITNESS VALUES AFTER 500 GENERATIONS AVERAGED OVER 10 RUNS

uncorrelated and restrictive capacity knapsacks						
n	MSs	MSd	PSb	PSv	PSw	PSr
100	*167*	**170**	161	160	155	166
250	850	**876**	852	842	810	846
500	*2550*	**2675**	2440	2513 3	2496	2426
weakly correlated and restrictive capacity knapsacks						
n	MSs	MSd	PSb	PSv	PSw	PSr
100	*236*	**242**	230	229	226	222
250	1066	**1134**	1046	1058	1098	1067
500	*3947*	**4071**	3719	3741	3713	3815
strongly correlated and restrictive capacity knapsacks						
n	MSs	MSd	PSb	PSv	PSw	PSr
100	405	**416**	403	405	389	408
250	2171	2204	2188	**2273**	2138	2205
500	*7028*	**7078**	6981	6883	6958	6946
uncorrelated and average capacity knapsacks						
n	MSs	MSd	PSb	PSv	PSw	PSr
100	225	**234**	218	231	212	231
250	1236	**1266**	1197	1208	1070	1263
500	4669	4697	4443	4674	3922	**4716**
weakly correlated and average capacity knapsacks						
n	MSs	MSd	PSb	PSv	PSw	PSr
100	295	**311**	290	294	292	304
250	*1520*	**1530**	1497	1491	1374	1519
500	5641	**5769**	5300	5525	4999	5710
strongly correlated and average capacity knapsacks						
n	MSs	MSd	PSb	PSv	PSw	PSr
100	476	**493**	479	483	470	**493**
250	2650	2716	2613	2610	2586	**2721**
500	10156	10285	10119	10131	10065	**10393**

In theory, the stopping criterion is that the algorithm halts once an optimal solution is found. It is taken for the convenience of analysing the first time of finding an optimal solution. If Φ_t includes an optimal solution, then assign

$$\Phi_t = \Phi_{t+1} = \Phi_{t+2} = \cdots$$

for ever. As a result, the population sequence $\{\Phi_t\}$ is formulated by a homogeneous Markov chain [17].

Since a state in the optimal set is always absorbing, so the transition matrix $\mathbf{P}$ can be written in the canonical form,

$$\mathbf{P} = \begin{pmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{R} & \mathbf{Q} \end{pmatrix}, \tag{12}$$

where $\mathbf{I}$ is a unit matrix, $\mathbf{O}$ a zero matrix and $\mathbf{Q}$ a matrix for transition probabilities among non-optimal populations. $\mathbf{R}$ denotes the transition probabilities from non-optimal populations to optimal populations.

Let $m(X)$ denote the expected number of generations needed to find an optimal solution when Φ_0 is at state X for the first time (thereafter it is abbreviated by the *expected hitting time*). Clearly for any initial population X in the optimal set, $m(X)$ is 0. Let $(X_1, X_2, \cdots)$ represent all populations in the non-optimal set and the vector $\vec{m}$ denote their expected number of generations respectively

$$\vec{m} = (m(X_1), m(X_2), \cdots)^T.$$

The following theorem [18, Theorem 11.5] shows that the expected hitting time can be calculated from the transition matrix.

Theorem 1 (Fundamental Matrix Theorem): The expected hitting time is given by

$$\vec{m} = (\mathbf{I} - \mathbf{Q})^{-1} \vec{1}, \tag{13}$$

where $\vec{1}$ is a vector all of whose entries are 1, the matrix $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}$ is called the *fundamental matrix*.

Two special values of the expected hitting time are often used to evaluate the performance of metaheuristics. The first value is the average of the expected hitting time, given by

$$\bar{m} = \frac{1}{\mid \mathcal{S} \mid} \sum_{X \in \mathcal{S}} m(X). \tag{14}$$

where $\mathcal{S}$ denotes the set of all populations. The average corresponds to the case when the initial population is chosen at random.

The second value is the maximum of the expected hitting time, given by

$$\max\{m(X); X \in \mathcal{S}\}, \tag{15}$$

The maximum corresponds to the case when the initial population is chosen at the worst state.

The population set $\mathcal{S}$ is divided into two parts: $\mathcal{S}_{\text{non}}$ denotes the set of all populations which don't include any optimal solution and $\mathcal{S}_{\text{opt}}$ the set of all populations which include at least one optimal solution.

Suppose the number of fitness evaluation is μ and then the runtime of a meta-heuristic equals $\mu m(X)$, denoted by $t(X)$.

B. Drift Analysis

In theory, the performance of metaheuristics is evaluated by the expected hitting time. Drift analysis is used for bounding the expected hitting time of metaheuristics [19]. In drift analysis, a *distance function* $d(X)$ is a non-negative function. Let $(X_1, X_2, \cdots)$ represent all populations in the non-optimal set and $\vec{d}$ denote the vector

$$(d(X_1), d(X_2), \cdots)^T.$$

Definition 2: Let $\mathbf{P}$ be the Markov chain associated with a metaheuristic algorithm and $d(X)$ be a distance function. For a non-optimal population X, the *drift* at state X is

$$\Delta(X) := d(X) - \sum_{Y \in \mathcal{S}_{\text{non}}} d(Y) P(X, Y).$$

The drift represents the one-step progress rate towards the global optima. Since the Markov chain $\{\Phi_t; t = 0, 1, \cdots\}$ is homogeneous, the above drift is independent of t.

The following theorem is a variant of the original drift theorem [17, Theorems 3 and 4].

Theorem 2 (Drift Analysis Theorem): (1) If the drift satisfies that $\Delta d(X) \geq 1$ for any state X, and $\Delta d(X) > 1$ for some state X, then the expected hitting time satisfies that $m(X) \leq d(X)$ for any initial population X, and $m(X) < d(X)$ for some initial population X.

(2) If the drift satisfies that $\Delta d(X) \leq 1$ for any state X, and $\Delta d(X) < 1$ for some state X, then the expected hitting time satisfies that $m(X) \geq d(X)$ for any initial population X, and $m(X) > d(X)$ for some initial population X.

Proof: We only prove the first conclusion. The second conclusion can be proven in a similar way.

The notation $\succ$ is introduced in the proof as follows: given two vectors $\vec{a} = [a_i]$ and $\vec{b} = [b_i]$, if for all i, $a_i \geq b_i$ and for some i, $a_i > b_i$, then write it by $\vec{a} \succ \vec{b}$. Similarly given two matrices $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$, if for all i, j, $a_{ij} \geq b_{ij}$ and for some pair i, j, $a_{ij} > b_{ij}$, then write it by $\mathbf{A} \succ \mathbf{B}$.

Let $\vec{1}$ denote the vector whose entries are 1, $\vec{0}$ the vector whose entries are 0 and $\mathbf{O}$ a matrix whose entries are 0. The condition of the theorem can be rewritten in an equivalent vector form:

$$\vec{d} - \mathbf{Q}\vec{d} = \vec{1} + \vec{e},$$

where $\vec{e} \succ \vec{0}$.

Then we have

$$\begin{aligned} &\vec{d} - \mathbf{Q}\vec{d} - \vec{1} - \vec{e} = \vec{0}, \\ &(\mathbf{I} - \mathbf{Q})^{-1}(\vec{d} - \mathbf{Q}\vec{d} - \vec{1} - \vec{e}) = \vec{0}, \\ &(\mathbf{I} - \mathbf{Q})^{-1}(\vec{d} - \mathbf{Q}\vec{d} - \vec{1}) = (\mathbf{I} - \mathbf{Q})^{-1}\vec{e}. \end{aligned}$$

Now let's bound the right-hand side. Since $\vec{e} \succ \vec{0}$, so entry $e_j > 0$ for some j. $\mathbf{P}$ is a transition matrix, $\mathbf{Q} \succ \mathbf{O}$ and the spectral radius of $\mathbf{Q}$ are less than 1, so $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1} \succ \mathbf{O}$. Since no eigenvalue of $\mathbf{N}$ is 0, for the j-column of $\mathbf{N}$, at least one entry is greater than 0 (otherwise 0 will be an eigenvalue of $\mathbf{N}$). Thus entry $n_{ij} > 0$ for some i. Then $n_{ij}e_j > 0$ and

$$(\mathbf{I} - \mathbf{Q})^{-1}\vec{e} \succ \vec{0}.$$

Hence we get

$$\begin{aligned} &(\mathbf{I} - \mathbf{Q})^{-1}(\vec{d} - \mathbf{Q}\vec{d} - \vec{1}) \succ \vec{0}, \\ &\vec{d} \succ (\mathbf{I} - \mathbf{Q})^{-1}\vec{1}. \end{aligned}$$

From Foundational Matrix Theorem, we know that

$$(\mathbf{I} - \mathbf{Q})^{-1}\vec{1} = \vec{m}.$$

Then we get $\vec{d} \succ \vec{m}$. This inequality implies the conclusion of the theorem. ■

The following consequence is directly derived from Fundamental Matrix Theorem.

Corollary 1: Let the distance function $d(X) = m(X)$, then the drift satisfies $\Delta(X) = 1$ for any state X in the non-optimal set.

Proof: From Fundamental Matrix Theorem: $(\mathbf{I} - \mathbf{Q})\vec{m} = \vec{1}$. Then we write it in the entry form and it gives $\Delta(X) = 1$ for any state X in the non-optimal set. ■

C. One Pure Strategy is Inferior or Equivalent to another Pure Strategy

In the subsection, we investigate the case that it is impossible to design a mixed strategy better than a pure strategy. Consider two metaheuristics algorithms: one using a pure strategy PS1 (PS1 for short) and another using a pure strategy PS2 (PS2 for short). Let $\vec{m}_{PS1}$ be the vector representing the expected hitting time of PS1 and the distance function $d(X) = m_{PS1}(X)$. For PS1, denote its corresponding drift by Δ_{PS1}:

$$\Delta_{PS1}(X) = d(X) - \sum_{Y \in S_{\text{non}}} P_{PS1}(X,Y)d(Y),$$

where $P_{PS1}(X,Y)$ represents the transition probability corresponding to PS1. According to Corollary 1, the drift $\Delta_{PS1}(X) = 1$ for all X in the non-optimal set.

For PS2, denote the corresponding drift by Δ_{PS1}:

$$\Delta_{PS2}(X) = d(X) - \sum_{Y \in S_{\text{non}}} P_{PS2}(X,Y)d(Y).$$

First we propose the "inferior" condition.

Definition 3: If the drift of PS1 and that of PS2 satisfy $\Delta_{PS1}(X) \geq \Delta_{PS2}(X)$ for any state X, and $\Delta_{PS1}(X) > \Delta_{PS2}(X)$ for some state X, then we call PS2 is *inferior* to PS1.

Design a *mixed strategy meta-heuristic derived from PS1 and PS2* (MS for short) at the population level: the probability of choosing a search strategy is the same for all individuals. When population Φ_t is at state X, one pure strategy is chosen from PS1 and PS2 based on a probability distribution. Denote the probability of choosing PS1 by $c_{PS2}(X)$ and the probability of choosing PS1 by $c_{PS2}(X)$. The sum $c_{PS1}(X) + c_{PS2}(X) = 1$.

Lemma 1: If PS2 is inferior to PS1, then for any mixed strategy metaheuristics derived from PS1 and PS2, the expected hitting time of MS satisfies that $m_{MS}(X) \geq m_{PS1}(X)$ for any initial population X, $m_{MS}(X) > m_{PS1}(X)$ for some state X.

Proof: Let $\Delta_{MS}(X)$ denote the drift associated with MS. For any state X, the drift of MS is

$$\begin{aligned}
\Delta_{MS}(X) =& d(X) - \sum_{Y \in S_{\text{non}}} P_{MS}(X,Y)d(Y) \\
=& c_{PS1}(X)[d(X) - \sum_{Y \in S_{\text{non}}} P_{PS1}(X,Y)d(Y)] \\
&+ c_{PS2}(X)[d(X) - \sum_{Y \in S_{\text{non}}} P_{PS2}(X,Y)d(Y)] \\
=& c_{PS1}(X)\Delta_{PS1}(X) + c_{PS2}(X)\Delta_{PS2}(X).
\end{aligned}$$

Since PS2 is inferior to PS1, we know that $\Delta_{PS1}(X) \geq \Delta_{PS2}(X)$ for any state X, and $\Delta_{PS1}(X) > \Delta_{PS2}(X)$ for some X. Therefore $\Delta_{PS1}(X) = 1 \geq \Delta_{MS}(X)$ for any state X, and $\Delta_{PS1}(X) = 1 > \Delta_{MS}(X)$ for some state X.

Applying Drift Analysis Theorem, we get the conclusion: the expected hitting time satisfies that $m_{MS}(X) \geq m_{PS1}(X)$ for any initial population X, and $m_{MS}(X) > m_{PS1}(X)$ for some initial population X. ■

From the above lemma, we infer two corollaries.

Corollary 2: If PS2 is inferior to PS1, then for any mixed strategy MS derived from PS1 and PS2, its average of the expected hitting time is greater than that of PS1.

Proof: According to the above lemma, the expected hitting time satisfies that $m_{MS}(X) \geq m_{PS1}(X)$ for any initial population X, and $m_{MS}(X) > m_{PS1}(X)$ for some initial population X. From the definition of average,

$$\bar{m} = \frac{1}{\mid \mathcal{S} \mid} \sum_{X \in \mathcal{S}} m(X),$$

then we get $\bar{m}_{MS} > \bar{m}_{PS1}$. ■

Corollary 3: If PS2 is inferior to PS1, then for any mixed strategy MS derived from PS1 and PS2, its maximum of the expected hitting time is not less than that of PS1.

Proof: According to the above lemma, the expected hitting time satisfies that $m_{MS}(X) \geq m_{PS1}(X)$ for any initial population X. Then we get

$$\max\{m_{MS}(X); X \in \mathcal{S}\} \geq \max\{m_{PS1}(X); X \in \mathcal{S}\}$$

and prove the conclusion. ■

Next we propose the "equivalent" condition.

Definition 4: If the drift of PS1 and that of PS2 satisfy $\Delta_{PS1}(X) = \Delta_{PS2}(X)$ for any state X, then we call PS1 is *equivalent* to PS2.

The following lemma is direct corollary of Drift Analysis Theorem.

Lemma 2: If PS2 is equivalent to PS1, then for any mixed strategy MS derived from PS1 and PS2, its the expected hitting time satisfies that $m_{MS}(X) = m_{PS1}(X)$ for any initial population X.

D. One Pure Strategy is Complementary to Another Pure Strategy

In the subsection, we investigate the case that it is possible to design a mixed strategy better than a pure strategy. We propose the "complementary" condition. Like the previous subsection, the distance function $d(X) = m_{PS1}(X)$.

Definition 5: If the drift of PS1 and that of PS2 satisfy $\Delta_{PS1}(X) < \Delta_{PS2}(X)$ for some state X, then we call PS2 is *complementary* to PS1.

Lemma 3: If PS2 is complementary to PS1, then there exists a mixed strategy MS derived from PS1 and PS2, and its the expected hitting time satisfies that $m_{MS}(X) \leq m_{PS1}(X)$ for any initial population X, and $m_{MS}(X) < m_{PS1}(X)$ for some initial population X.

Proof: First we construct a mixed strategy derived from PS1 and PS2. The construction follows a well-known principle: at one state, if a pure strategy has a better performance than the other at a state, then the strategy should be applied with a higher probability at that state.

1) When Φ_t is at state X, if the drift $\Delta_{PS1}(X)$ is greater than the drift $\Delta_{PS2}(X)$, then the probability of choosing PS1 is set to 1, that is, $c_{PS1}(X) = 1$.
2) When Φ_t is at state X, if the drift $\Delta_{PS1}(X)$ equals to the drift $\Delta_{PS2}(X)$, then the probability of choosing PS1 is set to any value between $[0, 1]$, that is, $0 \leq c_{PS1}(X) \leq 1$.
3) Since PS2 is complementary to PS1, so there exists one state X such that the drift $\Delta_{PS2}(X)$ is larger than the drift $\Delta_{PS1}(X)$. When Φ_t is at such a state X, then the probability of choosing PS2 is set to any value greater than 0, that is, $0 < c_{PS2}(X) \leq 1$.

In this way a mixed strategy MS is constructed from PS1 and PS2.

Next we bound the drift of the mixed strategy. For any state X, the drift of the mixed strategy is

$$\begin{aligned}\Delta_{MS}(X) =& d(X) - \sum_{Y \in S_{\text{non}}} P_{MS}(X,Y)d(Y) \\ =& c_{PS1}(X)[d(X) - \sum_{Y \in S_{\text{non}}} P_{PS1}(X,Y)d(Y)] \\ &+ c_{PS2}(X)[d(X) - \sum_{Y \in S_{\text{non}}} P_{PS2}(X,Y)d(Y)] \\ =& c_{PS1}(X)\Delta_{PS1}(X) + c_{PS2}(X)\Delta_{PS2}(X).\end{aligned}$$

Based on the construction of the mixed strategy, the analysis of the drift is classified into three cases.

1) $\Delta_{PS1}(X) > \Delta_{PS2}(X)$*:* in this case, the probability of choosing PS1 is 1, that is, $c_{PS1}(X) = 1$. Thus the drift satisfies: $\Delta_{MS}(X) = \Delta_{PS1}(X)$.
2) $\Delta_{PS1}(X) = \Delta_{PS2}(X)$*:* in this case, the drift satisfies: $\Delta_{MS}(X) = \Delta_{PS1}(X)$.
3) $\Delta_{PS1}(X) < \Delta_{PS2}(X)$*:* in this case, the probability of choosing PS2 is greater than 0, that is, $c_{PS2}(X) > 0$. Thus the drift satisfies: $\Delta_{MS}(X) < \Delta_{PS1}(X)$.

Summarising all three cases, we see that the drift of the mixed strategy satisfies: $\Delta_{MS}(X) \geq \Delta_{PS1}(X) = 1$ for any state X, and $\Delta_{MS}(X) > \Delta_{PS1}(X) = 1$ for some state X.

Finally applying Drift Analysis Theorem, we come to the conclusion: the expected hitting time satisfies: $m_{MS}(X) \leq m_{PS1}(X)$ for any initial population X, and $m_{MS}(X) < m_{PS1}(X)$ for some initial population X. ■

From the above lemma, we easily draw two results about the average and maximum of the expected hitting time.

Corollary 4: If PS2 is complementary to PS1, then there exists a mixed strategy MS derived from PS1 and PS2 and its average of the expected hitting time is less than that PS1.

Proof: According to the above lemma, the expected hitting time satisfies: $m_{MS}(X) \leq m_{PS1}(X)$ for any initial population X, and $m_{MS}(X) < m_{PS1}(X)$ for some initial population X. From the definition

$$\bar{m} = \frac{1}{\mid \mathcal{S} \mid} \sum_{X \in \mathcal{S}} m(X),$$

then we get $\bar{m}_{MS} < \bar{m}_{PS1}$. ■

Corollary 5: If PS2 is complementary to PS1, then there exists a mixed strategy MS derived from PS1 and PS2 and its maximum of the expected hitting time is no more than that PS1.

Proof: According to the above lemma, the expected hitting time satisfies: $m_{MS}(X) \leq m_{PS1}(X)$ for any initial population X. Then we get

$$\max\{m_{MS}(X); X \in \mathcal{S}\} \leq \max\{m_{PS1}(X); X \in \mathcal{S}\}$$

and prove the conclusion. ■

E. Complementary Strategy Theorem

Combining Lemmas 1, 2 and 3 together, we get the main result about mixed strategy metaheuristics. It gives an answer to the question: under what condition, mixed strategy metaheuristics may outperform pure strategy metaheuristics.

Theorem 3 (Complementary Strategy Theorem): Consider two metaheuristics algorithms: one using pure strategy PS1 and another using pure strategy PS2. The condition of PS2 being complementary to PS1 is sufficient and necessary if there exists a mixed strategy MS derived from PS1 and PS2 and its the expected hitting time satisfies: $m_{MS}(X) \leq m_{PS1}(X)$ for any initial population X, and $m_{MS}(X) < m_{PS1}(X)$ for some initial population X.

Proof: Given PS1 and PS2, their relation is classified into exact three types: PS2 is inferior, or equivalent, or complementary to PS1. Thus combining Lemmas 1, 2 and 3 together, we get the conclusion. ■

The theorem can be explained intuitively as follows.

1) If one pure strategy is inferior to another pure strategy, then it is impossible to design a mixed strategy with a better performance. So mixed strategy metaheuristics doesn't always outperform pure strategy metaheuristics.
2) If one pure strategy is complementary to another one, then it possible to design a mixed strategy better than the pure strategy. But it does not mean all mixed strategies will outperform the pure strategy.
3) The construction of a better mixed strategy metaheuristics should follow a general principle: if using a pure strategy has a better progress rate (in terms of the drift) than that using the other at a state, then the strategy should be applied with a higher probability at that state. This principle is general, but the design of a better mixed strategy is strongly dependent on the problem.

For the average of the expected hitting time, we may obtain a similar consequence after combining Corollaries 2, 4 and Lemma 2 together.

Corollary 6: The condition of PS2 being complementary to PS1 is sufficient and necessary if there exists a mixed strategy MS derived from PS1 and PS2 and its average of the expected hitting time is less than than that of PS1.

But the sufficient and necessary condition for the maximum of the expected hitting time is more complex.

F. An Example

Consider an instance of the 0-1 knapsack problem: the value of items $v_1 = n$ and $v_i = 1$ for $i = 2, \cdots, n$, the weight of items $w_1 = n$ and $w_i = 1$ for $i = 2, \cdots, n$. The capacity $C = n$. The fitness function is

$$f(x) = \begin{cases} n, & \text{if } s_1 = 1, s_2 = \cdots s_n = 0, \\ \sum_{i=1}^{n} s_i, & \text{if } s_1 = 0, \\ \text{infeasible}, & \text{otherwise.} \end{cases} \tag{16}$$

For the four pure EAs given in the previous section, it is easy to verify that

1) PSr is equivalent to PSb,
2) PSw is inferior to PSb,
3) PSv is complementary to PSb.

IV. Conclusions

The main contribution of the paper is Complementary Strategy Theorem. From the theoretical viewpoint, the theorem provides an answer to the question: under what condition, mixed strategy metaheuristics may outperform pure strategy metaheuristics. The theorem asserts that given two meta-heuristic algorithms where one uses a pure strategy PS1 and the other uses a pure strategy PS2, the condition of PS2 being complementary to PS1 is sufficient and necessary if there exists a mixed strategy algorithm derived from PS1 and PS2 and its the expected hitting time satisfies: $m_{MS}(X) \leq m_{PS1}(X)$ for any initial population X, and $m_{MS}(X) < m_{PS1}(X)$ for some initial population X. To the best of our knowledge, no similar sufficient and necessary condition was rigorously established before. The theorem itself is very intuitive, but this is the first time to rigorously prove it with the help of drift analysis.

Besides the above theoretical analysis, experiments are also implemented. Experimental results demonstrate that mixed strategy EAs may outperform pure strategy EAs on the 0-1 knapsack problem in up to 77.8% instances. In the experiments, the performance of an EA is measured by the fitness function value of the archive after 500 generations.

There exists a great gap between empirical and theoretical studies. (1) In experiments, the optimal solution is usually unknown in most instances, then the expected hitting time is unavailable; (2) in theory, it is difficult to analyse the average fitness value of the archive after 500 generations or other fixed generations.

References

[1] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptual comparison,” *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, 2003.

[2] F. Glover and G. A. Kochenberger, *Handbook of Metaheuristics*. Springer, 2003.

[3] M. Gendreau and J.-Y. Potvin, *Handbook of Metaheuristics*. Springer, 2010.

[4] C. Blum, A. Roli, and M. Sampels, *Hybrid Metaheuristics: an Emerging Approach to Optimization*. Springer, 2008.

[5] E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg, “Hyper-heuristics: An emerging direction in modern search technology,” in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds. Springer, 2003, pp. 457–474.

[6] F. Neri, C. Cotta, and P. Moscato, *Handbook of memetic algorithms*. Springer, 2011, vol. 379.

[7] J. He and X. Yao, “A game-theoretic approach for designing mixed mutation strategies,” in *Proceedings of the 1st International Conference on Natural Computation, LNCS 3612*, L. Wang, K. Chen, and Y.-S. Ong, Eds. Changsha, China: Springer, August 2005, pp. 279–288.

[8] P. Dutta, *Strategies and Games: Theory and Practice*. MIT Press, 1999.

[9] H. Dong, J. He, H. Huang, and W. Hou, “Evolutionary programming using a mixed mutation strategy,” *Information Sciences*, vol. 177, no. 1, pp. 312–327, 2007.

[10] L. Shen and J. He, “A mixed strategy for evolutionary programming based on local fitness landscape,” in *Proceedings of 2010 IEEE Congress on Evolutionary Computation*. Barcelona, Spain: IEEE Press, 2010, pp. 350–357.

[11] J. He, F. He, and H. Dong, “Pure strategy or mixed strategy?” in *Evolutionary Computation in Combinatorial Optimization*, J.-K. Hao and M. Middendorf, Eds. Springer, 2012, pp. 218–229.

[12] P. K. Lehre and E. Özcan, “A runtime analysis of simple hyper-heuristics: To mix or not to mix operators,” in *Proceedings of FOGA 2013*, 2013.

[13] R. Varga, *Matrix Iterative Analysis*. Springer, 2009.

[14] S. Martello and P. Toth, *Knapsack Problems*. Chichester: John Wiley & Sons, 1990.

[15] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. New York: Springer Verlag, 1996.

[16] J. He and Y. Zhou, “A comparison of gas using penalizing infeasible solutions and repairing infeasible solutions on average capacity knapsack,” in *Advances in Computation and Intelligence*, L. Kang, Y. Liu, and S. Zeng, Eds. Springer, 2007, pp. 100–109.

[17] J. He and X. Yao, “Towards an analytic framework for analysing the computation time of evolutionary algorithms,” *Artificial Intelligence*, vol. 145, no. 1-2, pp. 59–97, 2003.

[18] C. Grinstead and J. Snell, *Introduction to Probability*. American Mathematical Society, 1997.

[19] J. He and X. Yao, “Drift analysis and average time complexity of evolutionary algorithms,” *Artificial Intelligence*, vol. 127, no. 1, pp. 57–85, 2001.