

# A Parallel Evolutionary Approach to Multi-objective Optimization

Xiang Feng<sup>†</sup> Francis C.M. Lau

*Department of Computer Science, The University of Hong Kong, Hong Kong*

**Abstract**—Evolutionary algorithms have been used since the mid-eighties to solve complex single and multi-objective optimization problems. More recently the swarm intelligent approaches such as particle swarm optimization and ant colony optimization have been successfully used for multi-objective optimization. This paper proposes a new approach based on the generic generalized particle model (GE-GPM) for computing in parallel approximate efficient solutions for the distribution problem with multiple objectives. Unlike the swarm optimization approaches, GE-GPM is inspired by physical models of particle dynamics. We use mathematical formulations to describe or predict the properties and evolution of different states of the particles. In particular, according to “differential equation theory”, we develop efficient optimization techniques for multi-objective problems. We also adopt methods of classical mechanics to tackle the problem of modeling the interaction among the particles. We show that GE-GPM, being inspired by classical mechanics, enables feasible multi-objective optimization in very large scales. The GE-GPM approach has a low computational complexity, which is crucial for the functioning of large-scale distribution problems.

**Keywords**—Evolutionary algorithm (EA), multi-objective optimization, swarm intelligence, generic generalized particle model (GE-GPM), kinematics and dynamics

## I. INTRODUCTION

While single-objective *evolutionary algorithms* (EAs) are well established and relatively easy to parallelize, this is not the case for multi-objective evolutionary algorithms. Until recently, multi-objective combinatorial optimization did not receive much attention in spite of its potential application. The reason is probably due to the difficulty of deriving multi-objective combinatorial optimization models that are satisfactory. Nevertheless, the need for parallelizing multi-objective evolutionary algorithms to solve multi-objective combinatorial optimization problems exists.

“Distribution problem” is a well-known fundamental combinatorial optimization problem. Distribution problems are much more difficult to solve than assignment problems. Many practical situations can be formulated as a distribution problem. In fact, assignment problems and transportation problems are both sub-problems of the distribution problem; a distributed problem can be much more complex.

In [1], [2], some methods were proposed to generate the entire set of exact solutions for simple assignment

problems. These methods however seemed to be efficient only for small-scale instances. For large-scale instances or more complex problems, their NP-hard complexity and the multi-objectivity make these problems “intractable” for those methods. For this reason, it makes sense to consider “approximate” methods such as swarm intelligent methods which have been found to be efficient in treating combinatorial optimization problems; these latter methods can function independently of the mathematical structure of the problem, and generate excellent solutions in a very short time [3].

This paper proposes a new approach—based on the *generic generalized particle model* (GE-GPM)—to compute in parallel approximate efficient solutions to the distribution problem with multiple objectives. By proposing the GE-GPM approach, we try to explore a potentially new branch of EA, which is based on the laws of physical mechanics. Just like other EAs which draw from observations of physical processes that occur in nature, the GE-GPM approach is inspired by physical models of particle kinematics and dynamics.

In GE-GPM, we use mathematical formulations to describe or predict the properties and the evolution of the different states of particles. The particles represent parameters in the problem which follow a path towards a solution. Borrowing from “differential equation theory”, we developed efficient techniques for solving multi-objective optimization problems. The goal: all objectives to be optimized individually and then collectively, and satisfying all the given restrictions.

In the physical world, mutual attraction between particles causes motion. The reaction of a particle to the field of potential would change the particle’s coordinates and energies. The change in the state of the particle is a result of the influence of the potential. For GE-GPM, the objectives of individual optimizations are reached by the autonomous self-driving forces of the particles. Global optimization is achieved by the potential of the field, and any restrictions of the problem are satisfied via interaction potential between the particles.

Each particle is described by some differential dynamic equations, and it moves (to a new state in the field) according to the results of these calculations. Specifically, each particle computes the effect of its autonomous self-driving force, the field potential and the interaction potential. If the particles cannot reach an equilibrium, they will proceed to

<sup>†</sup>Corresponding author: Xiang Feng, Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong; email:xfeng@cs.hku.hk.

execute a goal-satisfaction process.

Although there are obvious differences between particles in classical mechanics and those in GE-GPM, GE-GPM is by and large inspired by classical mechanics. GE-GPM enables feasible multi-objective optimization in very large scales. The approach has a low computational complexity, which is crucial for the functioning of large-scale distribution problems.

This paper is part of the authors' research work on distributed parallel theories and approaches for intelligent processing based on the generalized particle model (GPM). They proposed the crossbar composite spring net (CCSN) approach [4] from which the GPM approach has evolved. They studied distributed and parallel algorithms for intelligent processing based on GPM, and their application in networks. GPM's application in the bandwidth allocation problem was presented in [5]. Variations of the basic theme then resulted in several extended GPM models, including the "economic generalized particle model" (E-GPM) which draws upon the economics theory of Tatonnement processes; the model has also been applied to the bandwidth allocation problem.

All the authors' past methods based on GPM targeted at a specific application to a real-life problem. In this paper, the GE-GPM method is described as a "generic" method such that potentially it can be applied to a range of different problems.

The structure of this paper is as follows. In Section II, we present the multi-objective generic generalized particle model (GE-GPM). Section III introduces the parallel evolutionary GE-GPM algorithm for solving multi-objective problems. In Section IV, we discuss the physical meanings of GE-GPM. In Section V, we give some experimental results. We conclude the paper in Section VI.

## II. THE GE-GPM APPROACH FOR THE MULTI-OBJECTIVE DISTRIBUTION PROBLEM

**Definition 1:** In a multi-objective framework, the distribution problem can be formulated as

$$\left\{ \begin{array}{l} \min : z^q(X) = (C^q)^T X = \sum_{i=1}^I \sum_{j=1}^J c_{ij}^q x_{ij} \quad q = 1, 2, \dots, Q \\ s.t. \quad \sum_{i=1}^I x_{ij} = 1 \quad j = 1, 2, \dots, J \\ \sum_{j=1}^J x_{ij} = 1 \quad i = 1, 2, \dots, I \end{array} \right. \quad (1)$$

where  $X$  is a two-dimensional distribution vector, and  $C^q$  a two-dimensional weight vector ( $q = 1, 2, \dots, Q$ ).

With this problem model, we can now examine the evolutionary multi-objective model which can mathematically describe GE-GPM for the multi-objective distribution problem. The theory of evolution is a dynamical theory. The evolutionary dynamics will drive GE-GPM to the equilibrium state.

**Definition 2:** The distribution and weight dynamic equations of GE-GPM are defined, respectively, by

$$x(t+1) = x(t) + \Delta x(t) \quad (2)$$

$$c(t+1) = c(t) + \Delta c(t) \quad (3)$$

The two dynamic equations are seen as the "GE-GPM evolution" by fictitious agents (service particles and task particles), which manipulate the distribution and weight vectors until an equilibrium is reached. In GE-GPM, the rows and columns of distribution vector  $X$  are treated as two kinds of fictitious agents (service particles and task particles). In fact, the weight vector is invariable; the evolution of the weight vector only occurs in the computing process in order to obtain efficient solutions of the distribution vector.

For fictitious agents—service particles (O) and task particles (S), there are three factors related to the distribution vector ( $X$ ) and the weight vector ( $C$ ):

- personal utility ( $u$ ) (to realize the multiple objectives);
- minimal personal utility (to realize max-min fair distribution and to increase the whole utility) ( $F$ );
- interaction among particles (to satisfy the restrictions) ( $I$ ).

According to "differential equation theory", a variable's increment to make it minimum is equal to the sum of negative items from related factors differentiating the variable. So we have the following definitions.

**Definition 3:** The increments of distribution and weight are defined, respectively, by

$$\Delta x \approx \frac{dx}{dt} = - \sum_{q=1}^Q (\lambda_1^q \frac{\partial u_O^q}{\partial x} + \lambda_2^q \frac{\partial F_O^q}{\partial x}) - \lambda_3 \frac{\partial I_O}{\partial x} \quad (4)$$

$$\Delta c^q \approx \frac{dp}{dt} = -(\gamma_1^q \frac{\partial u_S^q}{\partial c} + \gamma_2^q \frac{\partial F_S^q}{\partial c}) - \gamma_3 \frac{\partial I_S}{\partial c} \quad q = 1, 2, \dots, Q \quad (5)$$

where  $\lambda_1^q, \lambda_2^q, \lambda_3, \gamma_1^q, \gamma_2^q, \gamma_3$  are coefficients ( $q = 1, 2, \dots, Q$ ).

**Definition 4:** Three kinds of factor functions for service particles and task particles are defined, respectively, by

$$u_{O_i}^q = 1 - \exp\left(-\sum_{j=1}^J c_{ij}^q \cdot x_{ij}\right) \quad q = 1, 2, \dots, Q \quad (6)$$

$$F_O^q = (k_O^q)^2 \ln \sum_{i=1}^I \exp[(u_{O_i}^q)^2 / 2(k_O^q)^2] \quad q = 1, 2, \dots, Q \quad (7)$$

$$I_O = a_1 \sum_{i=1}^I \left(\sum_{j=1}^J x_{ij} - 1\right)^2 + a_2 \sum_{j=1}^J \left(\sum_{i=1}^I x_{ij} - 1\right)^2 \quad (8)$$

$$u_{S_j}^q = 1 - \exp\left(-\sum_{i=1}^I c_{ij}^q \cdot x_{ij}\right) \quad q = 1, 2, \dots, Q \quad (9)$$

$$F_S^q = (k_S^q)^2 \ln \sum_{i=1}^J \exp[(u_{S_j}^q)^2 / 2(k_S^q)^2] \quad q = 1, 2, \dots, Q \quad (10)$$

$$I_S = I_O \quad (11)$$

where  $k_O^q, a_1, a_2, k_S^q$  are coefficients.

Now, we explain why the three kinds of functions are chosen.

- 1) The smaller the value of the summation in Eq. (6), the more profit the  $i$ th service particle gets. The optimization problem here is posed as a minimization problem. And we use the exponential function in order that

$u_{O_i}^q(t)$  would be between 0 and 1.  $u_{O_i}^q(t)$  can be regarded as the  $q$ -th dimensional utility of service particle. The smaller  $u_{O_i}^q(t)$  is, the more profit service particle  $O_i$  gets. Schematically, the  $q$ -th dimensional utility function  $u_{O_i}^q$  of a particle corresponds to the  $q$ -th dimensional coordinate of the  $q$ -th dimensional force-field. We define the distance from the bottom boundary to the upper boundary of all  $q$  dimensional force-fields to be 1. The physical meaning of GE-GPM is discussed in Section IV. A graphical presentation of  $u_{O_i}^q(t)$  is shown in Fig. 1. Obviously, the smaller  $u_{O_i}^q(t)$  is the better. The presentation of  $u_{S_j}^q(t)$  in Eq. (9) is similar.

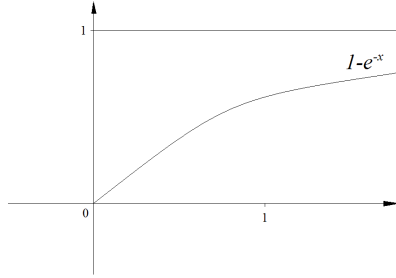


Fig. 1 Graphical presentation of  $u_{O_i}^q(t)$ .

- 2) For Eq. (7),  $0 < k_O^q < 1$  is a parameter to be tuned in the implementation. The smaller  $F_O^q$  is, the better. With Eq. (7), we attempt to construct a potential energy function,  $F_O^q$ , such that the decrease of its value would imply the decrease of the maximal utility of all the service particles. We prove that in Theorem 1. This way we can optimize the distribution problem in the sense that we consider not only the individual personal utility, but also the aggregate utilities, by decreasing the maximum utility of all the service particles again and again. In fact,  $k_O^q$  represents the strength of the downward gravitational force in the  $q$ -th dimensional force-field. The bigger  $k_O^q$  is, the faster the particles would move down; hence,  $k_O^q$  influences the convergence speed of the distribution problem.  $k_O^q$  needs to be carefully adjusted in order to minimize the  $q$ -th objective. Likewise, the explanation of  $F_S^q$  in Eq. (10) is similar.
- 3) For Eq. (8),  $0 < a_1, a_2 < 1$ . The smaller  $I_O$  is, the better.  $a_1, a_2$  are weights applied to the distribution availability of service and the satisfactory ratio of the demands, respectively. Eq. (8) describes the effect of interactions among service particles during the distribution process. The first term and the second term of  $I_O(t)$  perform penalty functions with respect to the constraints on the utilization of service (or resources) (i.e., service particles) and the degree of satisfaction of the demands (i.e., task particles) respectively. Therefore, distribution utilization and demands' satisfaction can be explicitly included as optimization objectives through some appropriate choices of the coefficients  $a_1$  and  $a_2$  respectively.

We presume that there are specific interactive forces among particles, and these forces may cause the potential energy components represented by the first and second term of  $I_O(t)$  to decrease. In Eq. (11),  $I_S$  is defined as the same as  $I_O$ .

We can therefore obtain the iteration velocity of service particles and task particles by the following equations, respectively.

$$v_{O_i}^q = du_{O_i}^q/dt = \frac{\partial u_{O_i}^q}{\partial x_{ij}} \frac{dx_{ij}}{dt} \quad (12)$$

$$v_{S_j}^q = du_{S_j}^q/dt = \frac{\partial u_{S_j}^q}{\partial c_{ij}^q} \frac{dc_{ij}^q}{dt} \quad (13)$$

$v_{O_i}^q$  represents the iteration velocity of the  $q$ -th objective by service particle  $O_i$ . Meanwhile,  $v_{O_i}^q$  represents the velocity of the downward movement of service particle  $O_i$  in the  $q$ -th dimensional force-field. The meaning of  $v_{S_j}^q$  is similar; it represents the iteration velocity of the  $q$ -th objective by task particle  $S_j$  and the velocity of the downward movement of task particle  $S_j$  in the  $q$ -th dimensional force-field.

**Theorem 1:** If  $k_O^q$  is very small, the decrease of  $F_O^q$  will cause a decrease of the service particles' maximal utility. (Likewise, if  $k_S^q$  is very small, a decrease of  $F_S^q$  will cause a decrease of the task particles' maximal utility.)

**Proof.** Supposing that

$$M(t) = \max_i [(u_{O_i}^q)^2(t)]. \text{ Because}$$

$$\begin{aligned} M(t) &= \max_i (u_{O_i}^q)^2(t) \leq \sum_{i=1}^I (u_{O_i}^q)^2(t) \\ &\leq I \cdot \max_i (u_{O_i}^q)^2(t) = I \cdot M(t), \end{aligned}$$

we have

$$\left[ e^{\frac{M(t)}{2(k_O^q)^2}} \right]^{2(k_O^q)^2} \leq \left[ \sum_{i=1}^I e^{\frac{(u_{O_i}^q)^2(t)}{2(k_O^q)^2}} \right]^{2(k_O^q)^2} \leq \left[ I \cdot e^{\frac{M(t)}{2(k_O^q)^2}} \right]^{2(k_O^q)^2}.$$

Simultaneously taking the logarithm of each side of the equation above leads to

$$M(t) \geq 2(k_O^q)^2 \ln \sum_{i=1}^I e^{\frac{(u_{O_i}^q)^2(t)}{2(k_O^q)^2}} \geq M(t) + 2(k_O^q)^2 \ln I,$$

$$2(k_O^q)^2 \ln \sum_{i=1}^I e^{\frac{(u_{O_i}^q)^2(t)}{2(k_O^q)^2}} \leq M(t)$$

$$\leq 2(k_O^q)^2 \ln \sum_{i=1}^I e^{\frac{(u_{O_i}^q)^2(t)}{2(k_O^q)^2}} - 2(k_O^q)^2 \ln I,$$

$$2F_O(t) \leq \max_i u_{O_i}^q(t) \leq 2F_O(t) - 2(k_O^q)^2 \ln I.$$

Since  $I$  is the number of service particles (the number of rows of the distribution vector  $X$ ),  $2(k_O^q)^2 \ln I$  is constant.

It turns out that  $F_O^q(t)$  at time  $t$  represents the maximum among  $u_{O_i}^q(t)$  obtained by the service particle  $O_i$ , namely, the minimum of the personal profit obtained by a service particle at time  $t$ . Hence decreasing  $F_O^q(t)$  implies the decrease of the maximal utility of the service particles.

**Definition 5:** (Max-min Fairness) [6] A feasible distribution  $X$  is max-min fair if and only if a decrease of any distribution  $x$  within the domain of feasible distributions must be at the cost of an increase of some already larger distribution  $x$ . Formally, for any other feasible distribution

$Y$ , if  $y_{ij} < x_{ij}$  then there must exist some  $i'$  such that  $x_{i'j} \geq x_{ij}$  and  $y_{i'j} > x_{i'j}$ .

**Theorem 2:** The behavior of the service particle  $O_i$  that is related to the term of Eq.(4),  $-\lambda_2^q \frac{\partial F_O^q}{\partial x}$ , will always bring about the decrease of the maximal utility of all service particles, and the decrement of the maximal utility is directly proportional to the coefficient vector  $\lambda_2^q$ . (Likewise, The behavior of the task particle  $S_j$  that is related to the term of the Eq.(5),  $-\gamma_2^q \frac{\partial F_S^q}{\partial c}$ , will always bring about the decrease of the maximal utility of all task particles, and the decrement of the maximal utility is directly proportional to the coefficient vector  $\gamma_2^q$ .)

**Theorem 3:** The behavior of the service particle  $O_i$  that is related to the term of the Eq.(4),  $-\lambda_1^q \frac{\partial u_{O_i}^q}{\partial x}$ , will always result in the decrease of the personal utility of service particle  $O_i$ , and the decrement of its personal utility is related to coefficient vectors  $\lambda_1^q$ . (Likewise, The behavior of the task particle  $S_j$  that is related to the term of the Eq.(5),  $-\gamma_1^q \frac{\partial u_{S_j}^q}{\partial c}$ , will always result in the decrease of the personal utility of task particle  $S_j$ , and the decrement of its personal utility is related to coefficient vectors  $\gamma_1^q$ .)

**Theorem 4:** The behavior of the service particle  $O_i$  that is related to the term of the Eq.(4),  $-\lambda_3 \frac{\partial I_O}{\partial x}$ , will decrease the potential interaction energy function  $I_O$ , with the intensity of the decrease being proportional to coefficient vector  $\lambda_3$ . (Likewise, The behavior of the task particle  $S_j$  that is related to the term of the Eq.(5),  $-\gamma_3 \frac{\partial I_S}{\partial c}$ , will decrease the potential interaction energy function  $I_S$ , with the intensity of the decrease being proportional to coefficient vector  $\gamma_3$ .)

**Theorem 5:** (Max-min fair allocation) Max-min fair allocation can be obtained by the mathematical model for the distribution problem with multi-objectives as defined in Eq.(1-13).

The proofs of Theorem 2-5 are omitted.

### III. THE PARALLEL GE-GPM ALGORITHM

The results given in the previous sections suggest that we may use a parallel implementation of the evolutionary generic generalized particle model approach to solve the multi-objective distribution problem. We consider the algorithm in Table 1 for this purpose.

The algorithm GE-GPM has in general a complexity of  $O(I + J)$ , where  $I + J$  is the number of particles (the sum of the number of rows and columns of  $X$ ). The time complexity of the algorithm is  $O(I_1)$ , where  $I_1$  is the number of iterations for Costep 2 (the while loop).

### IV. PHYSICAL MEANING OF GE-GPM

GE-GPM puts emphasis on

- providing a view of individual and whole optimization (with one to two objectives);
- parallelization with reasonably low time complexity;
- all objectives being optimized individually as well as collectively;
- the ability to deal with social interactions;
- the physical meaning of the model.

The mathematical model of GE-GPM has its physical meaning.

In GE-GPM, the rows and columns of the distribution vector  $X$  are treated as two kinds of generalized particles (service particles and task particles) that are located in two sets of force-fields, respectively, hence transforming the distribution problem into the kinematics and dynamics of the particles in the two sets of force-fields.

The two sets of force-fields are a set of service (or resource) force-fields and a set of task force-fields. Every force-field in a set of service force-fields or in a set of task force-fields is a  $Q$ -dimensional space where coordinates in the space are in  $[0, 1]$ .

If the number of minimum objectives is 1, the particles will move downwards on a 1-dimensional space (a line) ( $x_1 \in [0, 1]$ ) during the optimization process. If the number of minimum objectives is 2, the particles will move towards the origin in a 2-dimensional space (a plane) ( $x_1 \in [0, 1], x_2 \in [0, 1]$ ) during the optimization process. Analogously, if the number of minimum objectives is  $Q$ , the particles will move towards the origin on a  $Q$ -dimensional space ( $x_1 \in [0, 1], \dots, x_q \in [0, 1], \dots, x_Q \in [0, 1]$ ) during the optimization process, where  $x_q$  is a coordinate of the  $q$ -dimensional space.

Particles in GE-GPM move not only under outside forces, but also under their internal forces; hence they are different from particles in classical physics. The kinds of force-fields (resource force-field  $F_R$  and demands force-field  $F_D$ ) are geometrically independent, without any forces directly exerted from each other; they are mutually influenced and conditioned by each other through a reciprocal procedure whereby the distribution policy of the distributions  $x$  and the weight policy of the weights  $c$  change alternatively. In this way, the two sets of force-fields form a pair of reciprocal dual force-field sets.

In a resource force-field  $F_R$ , the coordinates of the  $Q$ -dimensional space of service particles represent the utilities of the rows of the distribution vector  $X$  that are described by the service particles. A particle will be influenced simultaneously by several kinds of forces in the  $Q$ -dimensional space, which include the gravitational force of the  $Q$ -dimensional space force-field where the particle is located, the pulling or pushing forces stemming from the interactions with other particles in the same force-field, and the particle's own autonomous driving force.

When the number of minimum objectives is 1, all the above-mentioned forces that are exerted on a particle are dealt with as forces along a vertical direction (along a line). Thus a particle will be driven by the resultant force of all the forces that act on it upwards or downwards, and moves along a vertical direction. The larger the downward resultant force on a particle, the faster the downward movement of the particle. When the downward resultant force on a particle is equal to zero, the particle will stop moving, being at an equilibrium status. As shown in Fig. 2, the service particles that have service or resource move in the resource force-fields  $F_R$ , and the task particles that require distribution move in the demand force-fields  $F_D$ .

**Table 1: GE-GPM algorithm**

1. Initialization:	
$t \leftarrow 0$	
$x_{ij}(t), c_{ij}^q(t)$	—Initialize in parallel
2. While ( $v_{O_i}^q \neq 0$ or $v_{S_j}^q \neq 0$ ) do	
$t \leftarrow t + 1$	
$u_{O_i}^q(t)$	—Compute in parallel according to Eq. (6)
$v_{O_i}^q(t)$	—Compute in parallel according to Eq. (12)
$u_{S_j}^q(t)$	—Compute in parallel according to Eq. (9)
$v_{S_j}^q(t)$	—Compute in parallel according to Eq. (13)
$dx_{ij}(t)/dt$	—Compute in parallel according to Eq. (4)
$x_{ij}(t) \leftarrow x_{ij}(t-1) + dx_{ij}(t)/dt$	—Compute in parallel according to Eq. (2)
$dc_{ij}^q(t)/dt$	—Compute in parallel according to Eq. (5)
$c_{ij}^q(t) \leftarrow c_{ij}^q(t-1) + dc_{ij}^q(t)/dt$	—Compute in parallel according to Eq. (3)

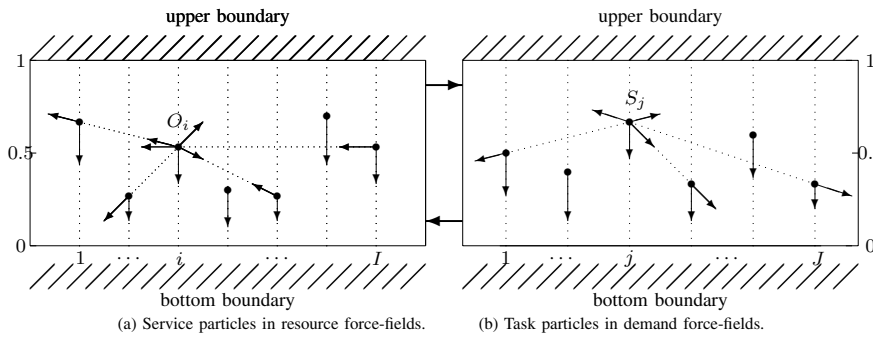


Fig. 2 The physical model of GE-GPM for the distribution problem with one objective.

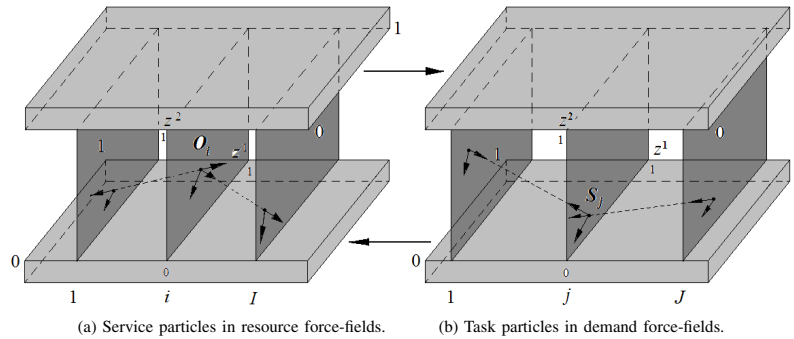


Fig. 3 The physical model of GE-GPM for the distribution problem with two objectives.

The downward gravitational force of a force-field on a particle causes a downward component of the motion of the particle, which represents the tendency that the particle pursues the common benefit of the whole group. The downward or upward component of the motion of a particle, which is related to the interactions with other particles, depends upon the strengths and categories of the interactions. The particle's own autonomous driving force is proportional to the degree the particle tries to move downwards in the force-field where it is located, i.e., the particle (service particle or task particle) tries to acquire its own minimum utility.

When the number of minimum objectives is 2, each service particle and task particle move towards the origin in a unit plane, as shown in Fig. 3.

When the number of minimum objectives is  $Q$ , each service particle and task particle move towards the origin

in a  $Q$ -dimensional space.

One major difference between the particle of the proposed generalized particle model and the particle of a classical physical model is that the generalized particle has its own driving force which depends upon the autonomy of the particle. All the generalized particles, both in different  $Q$ -dimensional spaces of the same force-field and in different force-fields simultaneously, evolve under their exerted forces; as long as they gradually reach their equilibrium positions from their initial positions which are set at random, we can obtain a feasible solution to the multi-objective distribution problem.

## V. SIMULATIONS

Here we give a distribution problem, and then use our method to find the solution.

$$C^1 = \begin{pmatrix} 8 & 7 & 10 & 1 & 4 \\ 7 & 11 & 16 & 0 & 5 \\ 2 & 7 & 6 & 19 & 15 \\ 3 & 6 & 4 & 7 & 11 \\ 14 & 5 & 7 & 3 & 2 \end{pmatrix}$$

$$C^2 = \begin{pmatrix} 10 & 7 & 10 & 5 & 11 \\ 3 & 19 & 5 & 3 & 13 \\ 2 & 18 & 9 & 0 & 1 \\ 13 & 3 & 7 & 5 & 12 \\ 7 & 4 & 6 & 15 & 3 \end{pmatrix} \quad q = 1, 2.$$

Find an  $X$  satisfying

$$\begin{cases} \min : z^q(X) = (C^q)^T X = \sum_{i=1}^5 \sum_{j=1}^5 c_{ij}^q x_{ij} & q = 1, 2 \\ \text{s.t.} & \sum_{i=1}^5 x_{ij} = 1 \quad j = 1, 2, \dots, 5 \\ & \sum_{j=1}^5 x_{ij} = 1 \quad i = 1, 2, \dots, 5 \end{cases}$$

We use the GE-GPM algorithm to solve this distribution problem.

**Step 1. Initialization:** ( $t = 0$ )

$$X = \begin{pmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{pmatrix}$$

$x_{ij}$  is also initialized as a random number between 0 and 1. Based on some experiments we have done, we found that the results are not affected by the initialization of  $X$ .

Standardization of  $C^1$  and  $C^2$ :

$$C^1 = \begin{pmatrix} 0.80 & 0.70 & 1.00 & 0.10 & 0.40 \\ 0.44 & 0.69 & 1.00 & 0 & 0.31 \\ 0.11 & 0.37 & 0.32 & 1.00 & 0.79 \\ 0.27 & 0.55 & 0.36 & 0.64 & 1.00 \\ 1.00 & 0.36 & 0.50 & 0.21 & 0.14 \end{pmatrix}$$

$$C^2 = \begin{pmatrix} 0.91 & 0.64 & 0.91 & 0.45 & 1.00 \\ 0.16 & 1.00 & 0.26 & 0.16 & 0.68 \\ 0.11 & 1.00 & 0.50 & 0 & 0.06 \\ 1.00 & 0.23 & 0.54 & 0.38 & 0.92 \\ 0.47 & 0.27 & 0.40 & 1.00 & 0.20 \end{pmatrix}$$

1. According to Eq. (1),  $z^q(X) = (C^q)^T X = \sum_{i=1}^5 \sum_{j=1}^5 c_{ij}^q x_{ij}$ , we get

$$z^1 = 2.6120, \quad z^2 = 2.6500$$

2. According to Eq. (6),  $u_{O_i}^q = 1 - \exp\left(-\sum_{j=1}^5 c_{ij}^q \cdot x_{ij}\right)$ , we compute in parallel and get

$$\begin{aligned} u_{O_1}^1 &= 0.4512 & u_{O_2}^1 &= 0.3861 & u_{O_3}^1 &= 0.4043 \\ u_{O_4}^1 &= 0.4311 & u_{O_5}^1 &= 0.3573 \\ u_{O_1}^2 &= 0.5425 & u_{O_2}^2 &= 0.3636 & u_{O_3}^2 &= 0.2839 \\ u_{O_4}^2 &= 0.4588 & u_{O_5}^2 &= 0.3737 \end{aligned}$$

3. According to Eq. (9),  $u_{S_j}^q = 1 - \exp\left(-\sum_{i=1}^5 c_{ij}^q \cdot x_{ij}\right)$ , we compute in parallel and get

$$\begin{aligned} u_{S_1}^1 &= 0.4079 & u_{S_2}^1 &= 0.4137 & u_{S_3}^1 &= 0.4706 \\ u_{S_4}^1 &= 0.3229 & u_{S_5}^1 &= 0.4102 \\ u_{S_1}^2 &= 0.4114 & u_{S_2}^2 &= 0.4663 & u_{S_3}^2 &= 0.4067 \\ u_{S_4}^2 &= 0.3283 & u_{S_5}^2 &= 0.4356 \end{aligned}$$

**Step 2. Compute in parallel:**

The first evolutionary iteration ( $t = 1$ ):

1. According to Eq. (4), we have

$$\begin{aligned} \Delta x_{ij} &\approx \frac{dx_{ij}}{dt} = -\sum_{q=1}^2 (\lambda_1^q \frac{\partial u_{O_i}^q}{\partial x_{ij}} + \lambda_2^q \frac{\partial F_O^q}{\partial x_{ij}}) - \lambda_3 \frac{\partial I_O}{\partial x_{ij}} \\ &= -\lambda_1^1 \frac{\partial u_{O_i}^1}{\partial x_{ij}} - \lambda_2^1 \frac{\partial F_O^1}{\partial x_{ij}} - \lambda_1^2 \frac{\partial u_{O_i}^2}{\partial x_{ij}} - \lambda_2^2 \frac{\partial F_O^2}{\partial x_{ij}} - \lambda_3 \frac{\partial I_O}{\partial x_{ij}} \end{aligned}$$

$$\text{where } \frac{\partial u_{O_i}^q}{\partial x_{ij}} = c_{ij}^q \cdot \exp\left(-\sum_{j=1}^5 c_{ij}^q \cdot x_{ij}\right)$$

$$\begin{aligned} \frac{\partial F_O^q}{\partial x_{ij}} &= \frac{\partial F_O^q}{\partial u_{O_i}^q} \cdot \frac{\partial u_{O_i}^q}{\partial x_{ij}} \\ &= (k_O^q)^2 \cdot \frac{\exp\{(u_{O_i}^q)^2 / [2(k_O^q)^2]\} \cdot [(u_{O_i}^q) / (k_O^q)]^2}{\sum_{i=1}^5 \exp\{(u_{O_i}^q)^2 / [2(k_O^q)^2]\}} \cdot \frac{\partial u_{O_i}^q}{\partial x_{ij}} \end{aligned}$$

$$\frac{\partial I_O}{\partial x_{ij}} = 2Ja_1 \sum_{i=1}^I \left( \sum_{j=1}^J x_{ij} - 1 \right) + 2Ia_2 \sum_{j=1}^J \left( \sum_{i=1}^I x_{ij} - 1 \right)$$

2. According to Eq. (5), we have

$$\Delta c_{ij}^q \approx \frac{dc_{ij}^q}{dt} = -(\gamma_1^q \frac{\partial u_{S_j}^q}{\partial c_{ij}^q} + \gamma_2^q \frac{\partial F_S^q}{\partial c_{ij}^q}) - \gamma_3 \frac{\partial I_S}{\partial c_{ij}^q}$$

$$\text{where } \frac{\partial u_{S_j}^q}{\partial c_{ij}^q} = x_{ij} \cdot \exp\left(-\sum_{i=1}^5 c_{ij}^q \cdot x_{ij}\right)$$

$$\begin{aligned} \frac{\partial F_S^q}{\partial c_{ij}^q} &= \frac{\partial F_S^q}{\partial u_{S_j}^q} \cdot \frac{\partial u_{S_j}^q}{\partial c_{ij}^q} \\ &= (k_S^q)^2 \cdot \frac{\exp\{(u_{S_j}^q)^2 / [2(k_S^q)^2]\} \cdot [(u_{S_j}^q) / (k_S^q)]^2}{\sum_{j=1}^5 \exp\{(u_{S_j}^q)^2 / [2(k_S^q)^2]\}} \cdot \frac{\partial u_{S_j}^q}{\partial c_{ij}^q} \end{aligned}$$

$$\frac{\partial I_S}{\partial c_{ij}^q} = 0$$

3. In addition,

$$\begin{aligned} x_{ij}(t=1) &= x_{ij}(t=0) + \Delta x_{ij}(t=1) \\ c_{ij}^1(t=1) &= c_{ij}^1(t=0) + \Delta c_{ij}^1(t=1) \\ c_{ij}^2(t=1) &= c_{ij}^2(t=0) + \Delta c_{ij}^2(t=1) \end{aligned}$$

$$\begin{aligned} \lambda_1^1 &= 0.05 & \lambda_2^1 &= 0.05 & \lambda_3^1 &= 0.05 & \lambda_2^2 &= 0.05 & \lambda_3 &= 0.01 \\ \gamma_1^1 &= 0.05 & \gamma_2^1 &= 0.05 & \gamma_1^2 &= 0.05 & \gamma_2^2 &= 0.05 & \gamma_3 &= 0.01 \end{aligned}$$

$$a_1 = 0.5 \quad a_2 = 0.5 \quad k_O^1 = k_O^2 = k_S^1 = k_S^2 = 0.8$$

As for these coefficients, we can draw the following conclusions from some experiments we have done.

- When  $k_O^q$  (or  $k_S^q$ ) is larger, the corresponding convergence speed is faster.
- If the values of  $\lambda$  and  $\gamma$  change in direct proportion, the experimental results will hardly be influenced.
- If we increase  $\lambda_1^q$ ,  $\lambda_2^q$  and do not touch the other coefficients, the  $q$ -th-objective will take precedence over all the other objectives.

We compute in parallel and get

$$X(t=1) = \begin{pmatrix} 0.1890 & 0.2007 & 0.1819 & 0.2277 & 0.2006 \\ 0.2133 & 0.1708 & 0.1881 & 0.2301 & 0.1978 \\ 0.2246 & 0.1766 & 0.2001 & 0.1968 & 0.2019 \\ 0.1973 & 0.2138 & 0.2100 & 0.2054 & 0.1736 \\ 0.1778 & 0.2109 & 0.2003 & 0.1886 & 0.2223 \end{pmatrix}$$

$$C^1(t=1) = \begin{pmatrix} 0.7751 & 0.6784 & 0.9719 & 0.0965 & 0.3876 \\ 0.4263 & 0.6687 & 0.9719 & 0 & 0.3004 \\ 0.1066 & 0.3586 & 0.3110 & 0.9648 & 0.7655 \\ 0.2616 & 0.5330 & 0.3499 & 0.6175 & 0.9690 \\ 0.9688 & 0.3489 & 0.4860 & 0.2026 & 0.1357 \end{pmatrix}$$

$$C^2(t=1) = \begin{pmatrix} 0.8818 & 0.6219 & 0.8816 & 0.4343 & 0.9702 \\ 0.1550 & 0.9717 & 0.2519 & 0.1544 & 0.6597 \\ 0.1066 & 0.9717 & 0.4844 & 0 & 0.0582 \\ 0.9690 & 0.2235 & 0.5231 & 0.3667 & 0.8926 \\ 0.4554 & 0.2624 & 0.3875 & 0.9650 & 0.1940 \end{pmatrix}$$

$$z^1(t=1) = 2.5243, \quad z^2(t=1) = 2.5590$$

Obviously,  $z^1(t=1) < z^1(t=0)$  and  $z^2(t=1) < z^2(t=0)$ , the distribution problem is optimized.

The evolutionary experimental results and the optimization trend from  $t=0$  to  $t=18$  are shown in Fig. 4 and Fig. 5.

The detailed evolutionary experimental results from  $t=2$  to  $t=18$  can be found in Appendix A.

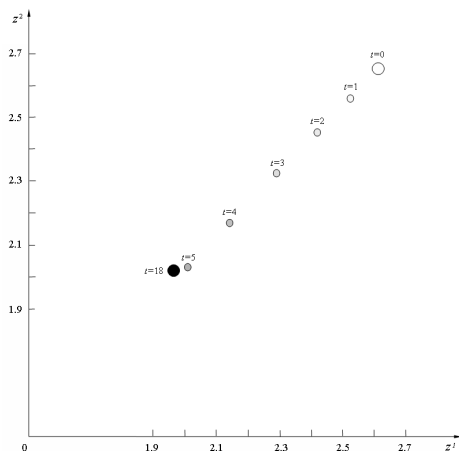


Fig. 4 Optimization from  $t=0$  to  $t=18$ .

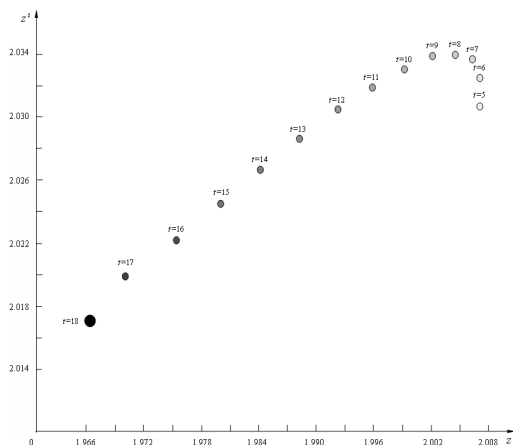


Fig. 5 Optimization from  $t=5$  to  $t=18$ .

As shown in Fig. 4 and Fig. 5, the two-objective distribution problem is optimized by the evolution of the GE-GPM algorithm. The convergence speed is faster at the beginning of evolution. The optimization trend of  $z^1$  and  $z^2$  reflects exactly the optimization of the problem, that is, the distribution problem is optimized step by step.

## VI. CONCLUSIONS

In this paper, we propose a novel evolutionary approach to solving multi-objective distribution problems, which is based on the generic generalized particle model (GE-GPM). The approach maps a given distribution problem to the movement of particles in a multi-dimensional space in a pair of (dual) sets of force-fields. The particles move according to certain rules defined by a mathematical model until arriving at a stable state; subsequently, the solution of the multi-objective distribution problem is obtained by anti-mapping the stable state.

Although there are many differences between particles in classical mechanics and those in GE-GPM, we have shown that being inspired by classical mechanics, GE-GPM enables feasible multi-objective optimization in very large scales. The GE-GPM approach has a low computational complexity, which is crucial for the functioning of large-scale distribution problems.

## Appendix A

The evolutionary experimental results from  $t=2$  to  $t=18$  are as follows.

$$X(t=2) = \begin{pmatrix} 0.1756 & 0.2015 & 0.1596 & 0.2620 & 0.2013 \\ 0.2292 & 0.1359 & 0.1738 & 0.2659 & 0.1952 \\ 0.2531 & 0.1493 & 0.2001 & 0.1934 & 0.2042 \\ 0.1940 & 0.2305 & 0.2219 & 0.2120 & 0.1416 \\ 0.1519 & 0.2236 & 0.2006 & 0.1755 & 0.2484 \end{pmatrix}$$

$$Z^1(t=2) = 2.4192$$

$$Z^2(t=2) = 2.4505$$

$$X(t=3) = \begin{pmatrix} 0.1590 & 0.2025 & 0.1319 & 0.3044 & 0.2022 \\ 0.2482 & 0.0942 & 0.1566 & 0.3088 & 0.1921 \\ 0.2860 & 0.1175 & 0.2000 & 0.1896 & 0.2068 \\ 0.1899 & 0.2506 & 0.2363 & 0.2201 & 0.1031 \\ 0.1214 & 0.2384 & 0.2009 & 0.1603 & 0.2789 \end{pmatrix}$$

$$Z^1(t=3) = 2.2931$$

$$Z^2(t=3) = 2.3211$$

$$X(t=4) = \begin{pmatrix} 0.1384 & 0.2037 & 0.0974 & 0.3572 & 0.2033 \\ 0.2710 & 0.0442 & 0.1360 & 0.3602 & 0.1885 \\ 0.3240 & 0.0807 & 0.1998 & 0.1856 & 0.2100 \\ 0.1848 & 0.2749 & 0.2537 & 0.2301 & 0.0564 \\ 0.0857 & 0.2558 & 0.2012 & 0.1426 & 0.3147 \end{pmatrix}$$

$$Z^1(t=4) = 2.1414$$

$$Z^2(t=4) = 2.1664$$

$$X(t=5) = \begin{pmatrix} 0.1194 & 0.2047 & 0.0655 & 0.4059 & 0.2045 \\ 0.2913 & 0 & 0.1176 & 0.4058 & 0.1852 \\ 0.3556 & 0.0497 & 0.1995 & 0.1824 & 0.2128 \\ 0.1803 & 0.2966 & 0.2690 & 0.2391 & 0.0150 \\ 0.0554 & 0.2705 & 0.2013 & 0.1278 & 0.3450 \end{pmatrix}$$

$$Z^1(t=5) = 2.0071$$

$$Z^2(t=5) = 2.0307$$

$$\begin{aligned}
X(t=6) &= \begin{pmatrix} 0.1184 & 0.2047 & 0.0633 & 0.4089 & 0.2046 \\ 0.2914 & 0 & 0.1174 & 0.4060 & 0.1853 \\ 0.3525 & 0.0522 & 0.1993 & 0.1832 & 0.2128 \\ 0.1801 & 0.2966 & 0.2689 & 0.2394 & 0.0150 \\ 0.0573 & 0.2695 & 0.2012 & 0.1290 & 0.3431 \end{pmatrix} & X(t=14) &= \begin{pmatrix} 0.1080 & 0.2062 & 0.0420 & 0.4390 & 0.2048 \\ 0.2925 & 0 & 0.1145 & 0.4077 & 0.1853 \\ 0.3414 & 0.0645 & 0.1987 & 0.1841 & 0.2114 \\ 0.1784 & 0.3016 & 0.2711 & 0.2421 & 0.0067 \\ 0.0646 & 0.2666 & 0.2002 & 0.1319 & 0.3367 \end{pmatrix} \\
Z^1(t=6) &= 2.0071 & Z^1(t=14) &= 1.9842 \\
Z^2(t=6) &= 2.0325 & Z^2(t=14) &= 2.0267 \\
\\
X(t=7) &= \begin{pmatrix} 0.1172 & 0.2048 & 0.0610 & 0.4122 & 0.2047 \\ 0.2915 & 0 & 0.1171 & 0.4061 & 0.1853 \\ 0.3500 & 0.0544 & 0.1991 & 0.1838 & 0.2127 \\ 0.1799 & 0.2969 & 0.2689 & 0.2397 & 0.0146 \\ 0.0589 & 0.2687 & 0.2010 & 0.1299 & 0.3415 \end{pmatrix} & X(t=15) &= \begin{pmatrix} 0.1065 & 0.2065 & 0.0392 & 0.4430 & 0.2048 \\ 0.2927 & 0 & 0.1142 & 0.4079 & 0.1853 \\ 0.3410 & 0.0654 & 0.1987 & 0.1838 & 0.2111 \\ 0.1782 & 0.3025 & 0.2716 & 0.2425 & 0.0052 \\ 0.0650 & 0.2666 & 0.2001 & 0.1317 & 0.3366 \end{pmatrix} \\
Z^1(t=7) &= 2.0062 & Z^1(t=15) &= 1.9799 \\
Z^2(t=7) &= 2.0336 & Z^2(t=15) &= 2.0245 \\
\\
X(t=8) &= \begin{pmatrix} 0.1160 & 0.2049 & 0.0586 & 0.4157 & 0.2048 \\ 0.2916 & 0 & 0.1168 & 0.4063 & 0.1853 \\ 0.3479 & 0.0564 & 0.1990 & 0.1842 & 0.2125 \\ 0.1797 & 0.2973 & 0.2690 & 0.2400 & 0.0140 \\ 0.0603 & 0.2681 & 0.2009 & 0.1307 & 0.3402 \end{pmatrix} & X(t=16) &= \begin{pmatrix} 0.1051 & 0.2068 & 0.0363 & 0.4471 & 0.2047 \\ 0.2928 & 0 & 0.1138 & 0.4081 & 0.1852 \\ 0.3406 & 0.0662 & 0.1987 & 0.1835 & 0.2109 \\ 0.1780 & 0.3035 & 0.2721 & 0.2428 & 0.0036 \\ 0.0653 & 0.2666 & 0.2000 & 0.1315 & 0.3366 \end{pmatrix} \\
Z^1(t=8) &= 2.0045 & Z^1(t=16) &= 1.9754 \\
Z^2(t=8) &= 2.0340 & Z^2(t=16) &= 2.0222 \\
\\
X(t=9) &= \begin{pmatrix} 0.1148 & 0.2050 & 0.0560 & 0.4193 & 0.2049 \\ 0.2917 & 0 & 0.1165 & 0.4065 & 0.1853 \\ 0.3462 & 0.0581 & 0.1989 & 0.1844 & 0.2124 \\ 0.1795 & 0.2978 & 0.2692 & 0.2403 & 0.0131 \\ 0.0614 & 0.2676 & 0.2007 & 0.1312 & 0.3391 \end{pmatrix} & X(t=17) &= \begin{pmatrix} 0.1037 & 0.2070 & 0.0334 & 0.4512 & 0.2046 \\ 0.2930 & 0 & 0.1134 & 0.4083 & 0.1852 \\ 0.3404 & 0.0670 & 0.1987 & 0.1832 & 0.2107 \\ 0.1778 & 0.3044 & 0.2726 & 0.2432 & 0.0020 \\ 0.0655 & 0.2667 & 0.2000 & 0.1313 & 0.3366 \end{pmatrix} \\
Z^1(t=9) &= 2.0021 & Z^1(t=17) &= 1.9709 \\
Z^2(t=9) &= 2.0338 & Z^2(t=17) &= 2.0198 \\
\\
X(t=10) &= \begin{pmatrix} 0.1135 & 0.2052 & 0.0533 & 0.4231 & 0.2049 \\ 0.2919 & 0 & 0.1161 & 0.4067 & 0.1853 \\ 0.3448 & 0.0597 & 0.1988 & 0.1845 & 0.2122 \\ 0.1793 & 0.2985 & 0.2695 & 0.2407 & 0.0121 \\ 0.0623 & 0.2672 & 0.2006 & 0.1316 & 0.3383 \end{pmatrix} & X(t=18) &= \begin{pmatrix} 0.1023 & 0.2073 & 0.0306 & 0.4552 & 0.2046 \\ 0.2931 & 0 & 0.1131 & 0.4085 & 0.1852 \\ 0.3402 & 0.0677 & 0.1988 & 0.1828 & 0.2105 \\ 0.1776 & 0.3053 & 0.2731 & 0.2435 & 0.0004 \\ 0.0657 & 0.2668 & 0.1999 & 0.1310 & 0.3367 \end{pmatrix} \\
Z^1(t=10) &= 1.9993 & Z^1(t=18) &= 1.9664 \\
Z^2(t=10) &= 2.0331 & Z^2(t=18) &= 2.0172 \\
\\
X(t=11) &= \begin{pmatrix} 0.1121 & 0.2054 & 0.0506 & 0.4270 & 0.2049 \\ 0.2920 & 0 & 0.1157 & 0.4070 & 0.1853 \\ 0.3436 & 0.0611 & 0.1987 & 0.1845 & 0.2120 \\ 0.1791 & 0.2992 & 0.2698 & 0.2410 & 0.0109 \\ 0.0631 & 0.2669 & 0.2005 & 0.1318 & 0.3377 \end{pmatrix} & u_{O_i}^1(t=18) &= ( 0.1752 & 0.1394 & 0.2105 & 0.2106 & 0.1558 ) \\
Z^1(t=11) &= 1.9959 & u_{O_i}^2(t=18) &= ( 0.2839 & 0.1267 & 0.1027 & 0.2171 & 0.1746 ) \\
Z^2(t=11) &= 2.0319 & u_{S_j}^1(t=18) &= ( 0.1637 & 0.1974 & 0.1880 & 0.1841 & 0.1601 ) \\
\\
X(t=12) &= \begin{pmatrix} 0.1108 & 0.2057 & 0.0478 & 0.4309 & 0.2049 \\ 0.2922 & 0 & 0.1153 & 0.4072 & 0.1853 \\ 0.3427 & 0.0623 & 0.1987 & 0.1844 & 0.2118 \\ 0.1789 & 0.2999 & 0.2702 & 0.2414 & 0.0096 \\ 0.0637 & 0.2668 & 0.2004 & 0.1319 & 0.3373 \end{pmatrix} & u_{S_j}^2(t=18) &= ( 0.1756 & 0.1552 & 0.1753 & 0.2221 & 0.1884 ) \\
Z^1(t=12) &= 1.9923 \\
Z^2(t=12) &= 2.0305 \\
\\
X(t=13) &= \begin{pmatrix} 0.1094 & 0.2059 & 0.0449 & 0.4349 & 0.2049 \\ 0.2924 & 0 & 0.1149 & 0.4074 & 0.1853 \\ 0.3420 & 0.0635 & 0.1987 & 0.1843 & 0.2116 \\ 0.1787 & 0.3008 & 0.2706 & 0.2418 & 0.0082 \\ 0.0642 & 0.2666 & 0.2003 & 0.1319 & 0.3369 \end{pmatrix} \\
Z^1(t=13) &= 1.9883 \\
Z^2(t=13) &= 2.0287
\end{aligned}$$

## REFERENCES

- [1] Ulungu EL, Teghem J. The two phases method: an efficient procedure to solve bi-objective combinatorial optimization problems. *Journal Foundations of Computing & Decision Sciences* 1995, 20(2):149-65.
- [2] Tuytens D, Teghem J, Fortemps Ph, Van Nieuwenhuysse K. Performance of the MOSA method for the bicriteria assignment problem. *Journal of Heuristics*, 2000, 6:295-310.
- [3] Pirlot M. General local search methods. *Europe Journal of Operational Research*, 1996, 92:493-511.
- [4] Dianxun Shuai, Xiang Feng. *Distributed Problem Solving in Multi-Agent Systems: A Spring Net Approach*. IEEE Intelligent Systems. 2005, 20(4): 66-74.
- [5] Dianxun Shuai, Xiang Feng. The Parallel Optimization of Network Bandwidth Allocation Based on Generalized Particle Model. *Computer Networks*. 2006, 50(9): 1219-1246.
- [6] Jean-yves LE Boudec. Rate adaptation, Congestion Control and Fairness: A Tutorial. November 22, 2005.