A Survey on Modeling and Optimizing Multi-Objective Systems

Jin-Hee Cho, *Senior Member, IEEE,* Yating Wang, Ing-Ray Chen, *Member, IEEE,* Kevin S. Chan, *Member, IEEE,* and Ananthram Swami, *Fellow, IEEE*

***Abstract*—Many systems or applications have been developed for distributed environments with the goal of attaining multiple objectives in the face of environmental challenges such as high dynamics/hostility, or severe resource constraints (e.g., energy or communications bandwidth). Often the multiple objectives are conflicting with each other, requiring optimal tradeoff analyses between the objectives. This work is mainly concerned with how to model multiple objectives of a system and how to optimize their performance. We first conduct a comprehensive survey of the state-of-the-art modeling and solution techniques to solve multi-objective optimization problems. In addition, we discuss pros and cons of each modeling and optimization technique for in-depth understanding. Further, we classify existing approaches based on the types of objectives and investigate main problem domains, critical tradeoffs, and key techniques used in each class. We discuss the overall trends of the existing techniques in terms of application domains, objectives, and techniques. Further, we discuss challenging issues based on the inherent nature of MOO problems. Finally, we suggest future work directions in terms of what critical design factors should be considered to design and analyze a system with multiple objectives.**

***Index Terms*—Multi-objective optimization, genetic algorithms, evolutionary algorithms, game theory, auction theory, trust, distributed systems.**

I. Introduction

REAL-world situations, such as those arising in economics or engineering environments, are complex and multi-dimensional in nature. Oftentimes, the scenarios are characterized by actors within these environments who are operating with a varied set of motivations and/or objectives [132]. Multiple objectives are often present, typically as a utility (or payoff) function. In many situations, however, maximizing all of the payoff functions is an over-constrained problem, as the objectives may be in conflict with each other. One can consider Pareto optimal situations where tradeoffs between these conflicting payoff functions are studied on a region, called the *Pareto frontier*. Navigation on the Pareto frontier enables one to optimize the design of such systems, performing multi-objective optimization (MOO) [46].

In this survey paper, we are particularly interested in how to model multiple objectives of a system, interwoven with complex system constraints (e.g., resource constraints, high adversarial conditions or dynamics, or distributed nature with no trusted centralized entity). Further, our interest is in how to optimize the performance of a system with multiple objectives.

Example applications include coalition formation (or team composition), cluster formation, task assignment, task scheduling, or resource allocation in various network environments including wireless sensor networks, mobile ad hoc networks, cloud computing, multi-agent systems, web-based social networks, supply chain environments, P2P networks, and so forth. Many MOO techniques have been explored, such as evolutionary algorithms, game theoretic approaches, and other metaheuristic algorithms. We aim to summarize the general trends on how modeling and solution techniques to solve MOO problems evolve as the main concerns of system platforms change.

For those who want to take a first step to initiate their research in the area of modeling and optimizing systems with multiple objectives, we are hopeful that this paper can provide useful background and guidelines.

A. Existing Survey Papers on Multi-Objective Optimization

Researchers have explored MOO problems since the 1970's in various domains for system control, decision making, circuit design, operations research, networking and telecommunications protocol design, and so forth. Several comprehensive survey papers on MOO solutions have appeared since the 1990's.

Shin and Ravindran [185] survey *interactive methods* to solve continuous MOO problems and corresponding applications. The authors discuss characteristics of preference assessments, assumptions to ensure the functionality of a method, and relationships between different methods. Ulungu and Teghem [202] review existing work on *multi-objective combinatorial optimization* (MOCO) problems because *multi-objective linear programming* (MOLP) methods had failed to solve MOO problems with discrete variables in many real-world applications.

With significantly increased attention on MOO problems and solution methods, many survey papers have been published from 2000 until now. In particular, several MOO survey papers discuss evolutionary algorithms [46, 48, 47, 56, 117, 198, 205] or bio-inspired algorithms [62, 170]. Okahe et al. [156] conduct a survey on how to measure quality of MOO algorithms and propose various types of performance indices. Also very recently Meng et al. [135] published a survey paper on MOO design methods using game theory. As seen in [135], a change of MOO modeling technique may be necessitated due

J-H. Cho, K.S. Chan and A. Swami are with US Army Research Laboratory Adelphi, MD email: {jin-hee.cho.civ, kevin.s.chan.civ, ananthram.swami.civ}@mail.mil

Y. Wang and I-R. Chen are with the Computer Science Department, Virginia Tech, Falls Church, VA email: {yatingw, irchen}@vt.edu

to diverse needs of distributed systems with distinct multiple objectives as opposed to centralized systems with only system level objectives.

Compared to the existing survey papers, our survey paper is unique in providing a comprehensive survey on both solution techniques (e.g., scalarization-based, metaheuristics, hybrid metaheuristics, and trust-based approaches) and modeling techniques (e.g., cooperative game theory or auction theory). We use our prior classification method developed in [43] with an in-depth survey to classify existing MOO modeling and solution techniques based on the types of objectives. Then we discuss the overall trends to analyze the tradeoff between solution optimality and solution search efficiency (i.e., complexity) of the surveyed MOO solution techniques. We summarize the overall trends of existing works to solve MOO problems, provide discussions of key challenging issues in MOO problems, and point to potential research directions based on the overall trends observed from the survey results.

B. Key Contributions

A preliminary version of this work was published in [43]. We substantially extend the conference paper [43] with the following contributions:

1) This work provides a comprehensive survey of not only key modeling techniques for modeling applications/systems with MOO requirements, but also solution techniques. We discuss each technique with its pros and cons that could lead to an insightful decision on the choice of techniques based on the distinct needs of each application;
2) This work uses the classification method, developed in our prior work [43], to categorize existing studies on various MOO problems in terms of the characteristics of objectives. This work substantially extends [43] with more detailed descriptions of techniques and trends observed based on the in-depth literature review associated with the three classes. Based on the classification method in [43], we classify existing works into three classes based on the nature of objectives, either system objectives or individual objectives in which an individual has its own goal as a member of a group which has a different goal to achieve;
3) We discuss the overall trends to analyze the tradeoff between solution optimality and solution search efficiency (i.e., complexity) observed in the surveyed MOO solution techniques. We provide the general trends of the application domains, objectives, and key techniques used for modeling and solving MOO problems. We provide insights on how modeling and solution techniques for MOO problems have evolved as emerging platforms have new needs to meet; and
4) Our work is the first that surveys trust-based solution techniques for solving MOO problems. We suggest future research directions based on trust-based heuristic approaches that are efficient and effective in achieving multiple, conflicting goals of a system under dynamic, hostile, and distributed environments.

C. Structure of this Survey Paper

The rest of this paper is structured as follows.

- Section II: We survey common applications to which MOO modeling and solution techniques are applicable. In addition, we discuss conflicting multiple objectives that are considered in example applications.
- Section III: We provide basic background knowledge about MOO problem definition and formulation. We also explain Pareto optimum and frontier for achieving MOO.
- Section VI: We survey solution techniques that solve various MOO problems, including scalarization-based methods, metaheuristics, hybrid metaheuristics, and trust-based solutions. We discuss the pros and cons of each solution technique.
- Section V: We survey modeling techniques for applications/systems with MOO goals, including cooperative game theory and auction theory. We discuss the pros and cons of each modeling technique.
- Section VI: We classify existing modeling and solution techniques to solve various MOO problems into three classes according to the types of objectives.
- Section VII: We discuss the challenging issues derived from the nature of MOO problems, including uncertainty, Pareto optimality conditions, duality, solvability, and stability, and the tradeoff between optimality and complexity.
- Section VIII: We suggest future research directions based on the overall trends observed from the survey results.
- Section IX: We summarize the key ideas from this survey.

II. Systems with Multiple Objectives

MOO problems are commonly encountered in many applications. In this section, we survey the major application domains that have explored techniques or algorithms to solve MOO problems.

Business Settings: Multiple criteria decision making situations are commonly observed where coalitions or alliances are established between buyers and sellers [99], customers and vendors [30], multiple business partners [96], and supply chain alliance in marketplaces [97]. The tradeoff issues investigated in [30, 96, 97] due to conflicting multiple objectives include: (1) maximization of sales by vendors vs. minimization of prices by customers [30]; (2) fair distribution of benefits by individual firms vs. overall maximum benefit based on the sum of all firms' benefit [96]; and (3) maximization of an individual's payoff vs. maximum profit of the supply chain alliance [97]. Overall we observe that conflicting interests exist between individuals and systems.

Tactical environments: Many military settings have multiple objectives in contexts such as task team formation [14], asset-to-task matching in tactical or surveillance systems [36, 49, 65], or team formation in multi-agent systems (e.g., robots, unmanned aerial or ground vehicles) [167, 174, 177, 180, 184]. The critical tradeoffs investigated in these works include: (1) agility vs. resource consumption [14]; (2) mission completion vs. fair resource utilization [36]; and (3) task cost (delay or communication overhead) vs. task completion (or coalition formation) [49, 65, 167, 174, 177, 180].

Web-based virtual environments: Many web-based virtual collaboration environments with service provision require team formation. The examples may include virtual collaboration networks requiring team formation among film actors or scientists where certain skill sets are required while some level of trust relationships are critical for cooperation [5]. Further, crowdsourcing in online environments is another good example of requiring task assignment with objectives including expertise, availability, or collaboration relationships [64]. Anagnostopoulos et al. [5] take a bi-criteria approximation approach to meet conflicting goals of minimizing communication cost while maximizing workload balance for task assignment in an online setting. Dorn et al. [64] solve a team composition problem based on member qualifications in terms of required skill sets and trust relationships with other members by using metaheuristics in an online environment. In these works [5, 64], the conflicting objectives are between skill sets and task related cost or fair workload.

Network resource allocation: Resource allocation techniques or algorithms have been developed for applications with multiple objectives. The examples with multiple but conficting objectives show the tradeoffs of conflicting objectives as follows: (1) team formation for service providers with the objectives of maximizing both load balance and quality-of-service (QoS) [194]; (2) primary and secondary user networks formation in cognitive networks with rate-interference tradeoff [15, 17, 18, 175]; (3) job formation to distribute the components of a program to suitable processors with conflicting objectives, maximizing throughput, reliability, or security with minimum communication cost [108, 122, 147, 193, 224]; and (4) cloud selection to achieve system functionalities based on the tradeoff between data management cost and QoS provided to users [103].

Most MOO problems including coalition formation (or team composition), task scheduling, task assignment, or resource allocation are usually combinatorial optimization problems, which are typically known to be *NP-complete* [8, 76]. Combinatorial optimization problems involve finding a best set from a finite set of objects. Many such problems cannot be solved in polynomial time.

III. Basic Concepts

A. Definition of MOO Problem

Optimization involves identifying the values of decision (or free) variables that generate the maximum or minimum of one or more objectives. In most engineering problems, there may exist multiple, conflicting objectives and solving the optimization problems is not trivial or may not be even feasible [168].

Osyczka [157] defines MOO as a problem that solves "a vector of decision variables" meeting constraints and optimizing a vector function where each element is an objective function.

Coello [46] provides a clear formulation of multi-objective optimization problems. Given m inequality constraints and p equality constraints, identify a vector, $\bar{x}_n^* = [x_1^*, x_2^*, \dots, x_n^*]^\top$, that optimizes

$$\bar{f}_k(\bar{x}_n) = [o_1(\bar{x}_n), o_2(\bar{x}_n), \dots, o_k(\bar{x}_n)]^\top \tag{1}$$

such that

$$\begin{aligned} g_i(\bar{x}_n) &\geq 0, \quad i = 1, 2, \dots, m \\ h_i(\bar{x}_n) &= 0, \quad i = 1, 2, \dots, p \end{aligned} \tag{2}$$

where $\bar{x}_n = [x_1, x_2, \dots, x_n]^\top$ is a vector of n decision variables. The constraints determine the "feasible region" F and any point $\bar{x}_n \in F$ gives a "feasible solution" where $g_i(\bar{x}_n)$ and $h_i(\bar{x}_n)$ are the constraints imposed on decision variables. The vector function $\bar{f}_k(\bar{x}_n)$ in (1) is a set of k objective functions, $o_i(\bar{x}_n)$ for $i = 1, \cdots, k$, representing k non-commensurable criteria [46].

B. Pareto Optimum

The concept of "Pareto optimality" is defined in [159]. A point $\bar{x}_n^* \in F$ is "Pareto optimal" (for minimization below) if the following holds for every $\bar{x}_n \in F$

$$\begin{gathered} \bar{f}_k(\bar{x}_n^*) \leq \bar{f}_k(\bar{x}_n) \\ \text{where } \bar{f}_k(\bar{x}_n) = [o_1(\bar{x}_n), o_2(\bar{x}_n), \dots, o_k(\bar{x}_n)]^\top, \\ \bar{f}_k(\bar{x}_n^*) = [o_1(\bar{x}_n^*), o_2(\bar{x}_n^*) \dots, o_k(\bar{x}_n^*)]^\top \end{gathered} \tag{3}$$

Pareto optimality gives a set of *nondominated* solutions. A feasible solution x is called "weakly nondominated" if there is no $y \in F$, such that $o_i(y) < o_i(x)$ for all $i = 1, 2, ..., k$. This means that there is no other feasible solution that can strictly dominate x. A feasible solution x is called "strongly nondominated" if there is no $y \in F$, such that $o_i(y) \leq o_i(x)$ for all $i = 1, 2, ...k$, and $o_i(y) < o_i(x)$ for at least one i. This means that there is no other feasible solution that can improve some objectives without worsening at least one other objective. If x is "strongly nondominated", it is also "weakly nondominated". However, the reverse does not hold [46].

A related concept is *Pareto efficiency* [159], which refers to a state in which resources cannot be reallocated to make any individual gain more without hurting any others' gain. Given an initial allocation, if we can achieve a different allocation making at least one individual better off without making any others worse off, then the starting state is called a *Pareto improvement*. An allocation is regarded as *Pareto efficient* (or Pareto optimal) when we cannot make any further Pareto improvements [33]. The standard Pareto optimality (or Pareto efficiency) mentioned above is often referred to as strong Pareto optimality (SPO). On the other hand, weak Pareto optimality (WPO, or weak Pareto efficiency) is a weaker version of SPO, indicating a state in which there is no other alternative way for resource reallocation to make any individual better off. If an allocation is SPO, it is a WPO; but not vice-versa. Therefore, we can identify strongly nondominated sets for solutions under the SPO allocation while finding weakly nondominated sets for solutions under the WPO allocation [33].

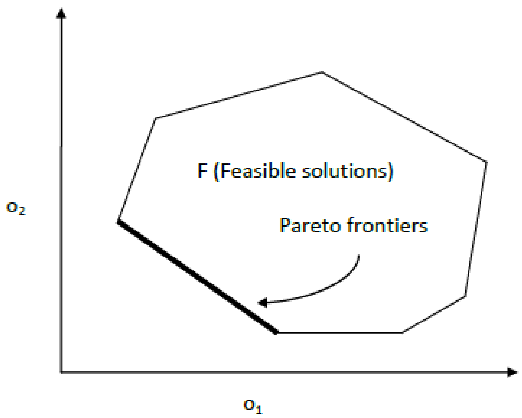


Fig. 1. An example MOO problem with objective minimization functions.

C. Pareto Frontier

A set (of feasible solutions) that is Pareto efficient is called the *Pareto frontier, Pareto set, or Pareto front*. The optimal solutions can be determined based on the tradeoffs within this set based on a designer's decisions for acceptable performance [33, 46].

Figure 1 shows an example of a MOO problem with two objective functions to be minimized [46]. The two goals are conflicting with each other as $o_1(\bar{x}_n)$ is minimized when $o_2(\bar{x}_n)$ is not, and vice-versa. The bold line in Figure 1 is the Pareto frontier or set. In general, the normal way to find optimal solution(s) is to obtain the Pareto set and then by comparing $\bar{f}_k(\bar{x}^*_n)$ for each solution $\bar{x}^*_n$ in the Pareto set, a designer can make a final decision for the optimal solution(s) [46].

If $\bar{x}^*_n \in F$ does not exist such that $\bar{f}_k(\bar{x}_n) < \bar{f}_k(\bar{x}^*_n)$, then we say $\bar{x}^*_n \in F$ is a *weakly nondominated* solution. This means that some solutions in $\bar{f}_k(\bar{x}_n)$ can achieve as much as $\bar{f}_k(\bar{x}^*_n)$ while other solutions in $\bar{f}_k(\bar{x}_n)$ can be strictly dominated by $\bar{f}_k(\bar{x}^*_n)$. If $\bar{x}^*_n \in F$ does not exist such that $\bar{f}_k(\bar{x}_n) \leq \bar{f}_k(\bar{x}^*_n)$ and for at least one value of k, $\bar{f}_k(\bar{x}_n) < \bar{f}_k(\bar{x}^*_n)$, then we call $\bar{x}^*_n \in F$ is a *strongly nondominated solution*. This means that solutions in $\bar{f}_k(\bar{x}^*_n)$ is strictly better than $\bar{f}_k(\bar{x}_n)$. Hence, if $\bar{x}^*_n$ is *strongly nondominated*, it is also *weakly nondominated*. However, the vice-versa does not hold [46].

IV. MOO Solution Techniques

In this section we discuss MOO solution techniques. We organize this section as follows: Section IV-A for scalarization-based techniques; Section IV-B for solution metaheuristics; Section IV-C on hybrid metaheuristics; and Section IV-D on trust-based approximation solutions.

A. Scalarization-based MOO Function Formulation

In this section, we discuss 'classical' MOO methods to formulate a single objective function that captures multiple objectives, mainly based on scalarization techniques. Table I summarizes the scalarization-based MOO formulation techniques along with an analysis of pros and cons.

1) Weighted Sum: A popular technique is the *weighted sum*, also called *scalarization*, that generates a single objective function by linearly combining multiple objective functions by

$$\text{Optimize } \bar{f}_k(x) = \textstyle\sum_{i=1}^{k} r_i o_i(x), \quad (4)$$
$$\text{where } 0 \leq r_i \leq 1, i = \{1, \ldots, k\}, \textstyle\sum_{i=1}^{k} r_i = 1$$

Here each weight represents the importance of that objective function [174, 176]. Deciding appropriate weights (i.e., the r_i's) for each objective is critical to obtaining good solutions. Since the different objectives, the o_i's, measure different quantities, appropriate normalization of units is also critical [46].

Pros and Cons: This method is computationally efficient in generating a *strong nondominated solution* [119] while not changing the structure of the problem by keeping the same set of constraints. However, the solution generated by this method heavily depends on the weight coefficients for combining the objectives. If the solution functions meeting the constraints for each objective are not linear, showing a concave tradeoff surface, a linear weighted sum cannot identify optimal solutions [73].

2) ε-Constraints: This technique helps generate a single objective function by picking one of the multiple objective functions, say the i-th, as the primary objective function and casting the remaining objective functions as constraints [62]. The objective function can be written as

$$\text{Optimize } o_i(x) \quad (5)$$
$$\text{subject to } o_j(x) \leq \varepsilon_j, j = 1, \ldots, k \text{ and } j \neq i$$

Matsatsinis and Delias [134] study task allocation problems in multi-agent decision making systems using *ε-constraints*. *Bi-criteria approximation* is a typical approach to solve a two-objective optimization problem, in which one of the objectives is interpreted as a constraint while the other objective is set as a system goal. Anagnostopoulos et al. [5] utilize bi-criteria approximation to optimize both communication overhead and workload balancing in an online task assignment problem.

Pros and Cons: The *ε-constraints* method leads to a weakly nondominated solution. However, if only one optimal solution exists, such a solution becomes strongly nondominated [46]. In this method, multiple rounds of searching for solutions using a different set of constraints can identify trade-off points among multiple objectives [171]. Its drawback is that it is a time-consuming process, and in the case with too many objectives, it is hard to formulate the problem itself [46]. The choice of ε significantly affects problem solutions when dealing with continuous objective values and also modifying the structure of the problem by adding extra constraints.

3) Goal Programming: This method is used by decision makers (DMs) to assign targeted goals that they want to attain for each objective. The objective function aims to minimize the absolute difference between the targeted goals and the achieved performance [54, 196], and is formulated as

$$\min \sum_{i=1}^{k} |o_i(x) - g_i| \quad (6)$$

where g_i refers to the target goal for objective i and X is a set of feasible solutions [54, 148].

TABLE I
SCALARIZATION-BASED MOO PROBLEM FORMULATION TECHNIQUES

Technique	Pros	Cons	Ref.
Weighted sum	Computationally efficient in generating a strong nondominated solution	Depending on weight coefficients, concave tradeoff curve may not be observed	[46, 119, 171, 174, 176]
ε-constraints	Can identify tradeoff points among multiple objectives by searching solutions using different set of constraints	Time-consuming process; difficult to formulate a problem with too many objectives; critical impact of ε on solutions	[5, 46, 62, 134, 171]
Goal programming	Computationally efficient if feasible solution space is found	Computationally inefficient if feasible solution space is not found	[45, 54, 67, 107, 148, 196]
Min-max	Results in the best possible optimal solution, given an equal priority for each objective	Computationally inefficient if feasible solution space is not found	[46, 192]
Elastic-constraints	Transforms some constraints to soft constraints for better solution search efficiency	Results in significant impact by a penalty factor (i.e., μ_k in Eq. (9))	[69, 113]
Weighted metric	Allows finding a set of Pareto optimal solutions of each objective if reference points as a vector of utopian objectives are known	Cannot solve non-differentiable MOO function	[138, 140, 228]
Achievement function	Allows finding Pareto optimal solutions by flexibly selecting reference points	Introduces complexity in selecting right reference points	[138]
Benson's method	Applicable in non-convex MOO problems when reference points are properly selected	Introduces extra constraints; not applicable for gradient-based methods for non-differentiable objective functions	[20, 56]
Utility (or value) function	Can be used with game theory to design a MOO problem related to resource allocation	Hard for an individual agent to have a global view in fully distributed systems	[56, 85]

Many variants of the goal programming method have been proposed. *Lexicographic goal programming* (or preemptive goal programming) is devised to specify priorities of goals. *Weighted goal programming* (or non-preemptive goal programming) examines direct tradeoffs between all deviational variables. *Chebyshev goal programming* seeks to minimize the maximal deviation from any goal, not the sum of all deviations. *Fuzzy goal programming* employs fuzzy set theory to estimate a level of deviations. For more details, readers are referred to [107].

Pros and Cons: This method is computationally efficient if the targeted goal is selected in the feasible solution space [67]. But a DM should set targeted goals appropriately in order to remove the characteristics of the problem that are not commensurable. This is not a trivial process unless information about the search space shape is known a priori. If the feasible region is not identifiable, this method is inefficient [45]. Further, a well-known weakness of this goal programming method is not generating solutions that are Pareto efficient [172]. Nevertheless, some techniques have been proposed to restore Pareto efficiency into goal programming while maintaining efficiency [197].

4) Min-Max Method: This aims to minimize the maximum deviations of the objective values from optimal objective values [192]. For example, we can formulate a min-max problem via

$$\min\left[\max Z_i(x)\right] \quad \text{for } i = 1, \ldots, k \tag{7}$$

where $Z_i(x)$ is obtained from the target value g_i by

$$Z_i(x) = \frac{|o_i(x) - g_i|}{g_i} \quad \text{for } i = 1, \ldots, k \tag{8}$$

Note that Eq. (7) differs from Eq. (6) in that Eq. (7) has a stricter standard than Eq. (6). Eq. (6) minimizes the sum of the differences between the optimal objective value and the goal objective value while Eq. (7) minimizes the difference per objective. Eq. (6) allows deviations of some objectives when other objectives are almost strictly met. In Eq. (7), the attainment of meeting each goal is checked per objective.

Pros and Cons: This technique can yield the best achievable optimal solution where all objectives have an equal priority for optimization [192]. If different priority levels are given, this formulation can be modified by using a "lexicographic goal programming technique," introducing a demand-level vector [192]. Priority levels can be considered by transforming Eq. (7) above into $\min[\max w_i Z_i(x)]$ that can generate either weak or strong Pareto optimal solutions [232]. However, if the feasible region is not identifiable, it is not computationally efficient [46]. Further, the normalization in Eq. (8) takes care of the unit normalization problem discussed earlier.

5) Elastic-Constraints Method: It was developed in order to incorporate benefits of the weighted sum and ε-constraints method while avoiding their weaknesses. This method is devised to solve a single objective as the benefit of a weighted sum while considering all efficient solutions like the ε-constraints method [69]. The underlying idea is that elastic constraints allow a problem to be solved easily by setting upper bounds on violated objective values which are used to penalize the constraint violation [69, 113]. A MOO function based on this elastic-constraints method is formulated by

$$\begin{aligned} &\min\nolimits_{\mathbf{x}\in S} \; [c_j^T \mathbf{x} + \sum_{k\neq j} \mu_k s_k] \\ &\text{subject to } \; c_k^T \mathbf{x} + l_k - s_k = \epsilon_k, \;\; k \neq j \\ &\qquad s_k, l_k \geq 0, k \neq j, \mathbf{x} \in S \end{aligned} \tag{9}$$

where ϵ_k is a constraint and μ_k is the penalty coefficient for a given objective k. This method uses two sets of variables including slack variables, l_k, and surplus variables, s_k, in order to transform the upper bounds on objective values into equality constraints for any $\mathbf{x} \in S$ (i.e., a set of feasible solutions) based an appropriate selection of s and l [69].

Pros and Cons: Wierzbicki et al. [218] discuss how the elastic-constraints method can be used in real world MOO problems. Since some constraints are hard (e.g., physical constraints) while other constraints can be soft (e.g., budget constraints), some objectives can be transformed as soft constraints by using the elastic-constraints method which increases efficiency in solution search [218]. Based on the experiments conducted by Ehrgott and Ryan [70], solution search complexity of the elastic-constraints method is significantly affected by the value of μ_k in which small μ_k incurs more computational complexity. Accordingly, the existence of Pareto optimality is strikingly affected by the value of μ_k.

6) Weighted Metric Method: This method uses a global criterion by minimizing the distance between a certain reference point, z_i^*, and a feasible objective region where each metric can be weighted with a different degree [138]. This weighted method is also called *compromise programming* [228]. A MOO function based on the weighted metric method can be formulated for $1 \leq p < \infty$ and a set of feasible solutions S by

$$\min\Big(\sum_{i=1}^{k} w_i(f_i(\mathbf{x}) - z_i^*)^p\Big)^{1/p} \quad (10)$$
$$\text{subject to } \mathbf{x} \in S$$

Assuming that the global ideal objective vector is known, a related reformulation based on a weighted Chebyshev problem [138] is given by

$$\min \ \max_{i=1,\cdots,k}[w_i(f_i(\mathbf{x}) - z_i^*)] \quad (11)$$
$$\text{subject to } \mathbf{x} \in S$$

Eq. (10) is Pareto optimal if either the solution is unique or $w_i > 0$ while Eq. (11) gives a weakly Pareto optimal solution for $w_i > 0$ [138, 140].

Pros and Cons: If reference points $\mathbf{z}^*$, a vector of utopian objective values, are known, Eq. (11) can identify a Pareto optimal solution of each objective. However, Eq. (10) is not differentiable which precludes the use of single objective optimizers using gradient information [138].

7) Achievement Function Method: This method is similar to the weighted metric method. However, it is different in that it does not fix a reference point as an ideal or uptopian objective vector and does not use a distance metric. In this sense, this method is a type of goal programming method aiming to minimize the deviation from ideal performance. These features can provide Pareto optimal solutions regardless of what reference point is selected [138]. It can formalize a MOO function as

$$\min \ \max_{i=1,\cdots,k}[w_i(f_i(\mathbf{x} - z_i^*)] + \rho\sum_{i=1}^{k}(f_i(\mathbf{x} - z_i^*) \quad (12)$$
$$\text{subject to } \mathbf{x} \in S$$

where w_i's are the normalized weights and $\rho > 0$.

Pros and Cons: As discussed above, due to the unique features of flexibly selecting a reference point, this method can find Pareto optimal solutions. However, this can also introduce complexity in identifying proper reference points.

8) Benson's Method: This method is similar to ε-constraints method in choosing a single objective while other objectives are considered as constraints. Also similar to the weighted metric method, it uses a vector of reference points, $\mathbf{z}^*$, but they are selected randomly from a feasible solution space. This method aims to maximize the sum of the non-negative differences between the reference point and the feasible solutions for each objective [20, 56]. It can be formulated by

$$\max\sum_{i=1}^{k}\max_{x\in S}[0, (z_i^* - f_i(\mathbf{x}))] \quad (13)$$
$$\text{subject to}$$
$$[f_1(\mathbf{x}), \cdots, f_k(\mathbf{x})] \leq \mathbf{z}^*$$
$$[g_1(\mathbf{x}), \cdots, g_m(\mathbf{x})] \geq 0$$
$$[h_1(\mathbf{x}), \cdots, h_j(\mathbf{x})] = 0, \mathbf{x} \in S$$

Pros and Cons: To avoid scaling problems, each individual difference should be normalized based on the same scale before the summation. If the random reference points, $\mathbf{z}^*$, are properly selected, this method can be also applied in non-convex multi-objective problems. However, this method introduces extra constraints. In addition, since the objective function is not differentiable, direct gradient based methods cannot be used [56].

9) Utility Function Method: This method can be used in a situation in which multiple users have individual objectives and associated utility functions, $\mathbf{U}$, where utility function u_k for user k should be valid over the feasible solution space. Solutions can be compared based on the values of utilities. For example, for two solutions i and j, solution i is preferred to solution j when $U(f(\mathbf{x}_i)) > U(f(\mathbf{x}_j))$ [56, 85]. This utility function method can formulate a MOO function by

$$\max \ U(f(\mathbf{x})) \quad (14)$$
$$\text{subject to}$$
$$[f_1(\mathbf{x}), \cdots, f_k(\mathbf{x})] \leq \mathbf{z}^*$$
$$[g_1(\mathbf{x}), \cdots, g_m(\mathbf{x})] \geq 0, \mathbf{x} \in S$$

where $f(\mathbf{x}) = [f_1(\mathbf{x}), \cdots, f_k(\mathbf{x})]^T$.

Pros and Cons: The underlying idea is simple to apply in contexts where multiple users have their own goals to maximize particularly using game theoretic approaches. If appropriate utility functions are given, it can be useful to solve diverse resource allocation problems. However, it is challenging to formulate utility functions which can be globally applicable (e.g., a system goal perspective) because there is no guarantee for an individual agent to obtain an global view efficiently in fully distributed environments.

B. Metaheuristics

In this section, we discuss the following well-known metaheuristics to solve MOO problems: (1) evolutionary algorithms and its variants; (2) ant colony optimization method; (3) particle swarm optimization method; (4) simulated annealing; (5) Tabu search; and (6) variable neighborhood search. Table

TABLE II
METAHEURISTICS METHODS

Technique	Pros	Cons	Ref.
Evolutionary Algorithms (EA)	Provides heuristic but close-to-optimal solutions	Computationally expensive fitness functions that often generate local optima	[205]
Nondominated Sorting Genetic Algorithm (NSGA)	Preserves solution diversity based on the crowding distance comparison process; High efficiency by preserving optimal solutions based on elitism	Poor performance under the failure of generating crowd solutions under multiple objectives	[47, 120]
Strength Pareto Evolutionary Algorithm (SPEA)	High solution quality maintained based on elitism with high solution diversity	Most computational overhead incurred in density estimation	[187]
Multi-Objective Quantum-inspired Evolutionary Algorithm (MQEA)	High scalability based on quantum-based mutation and crossover; high proximity to Pareto optimality; high solution diversity of nondominated solutions	Computationally intensive to reach convergence to Pareto front	[126]
Hierarchical Evolutionary Algorithm (HEA)	High efficiency in eliminating invalid or dominating solutions	Computationally intensive when the feasible solution space is large	[61]
Ant Colony Optimization (ACO)	Useful for dynamic applications	Not predictable for solution convergence time	[24, 63, 65]
Particle Swarm Optimization (PSO)	High quality solution with less time	May be difficult to identify the optimal parameters in some cases	[104, 109, 123]
Simulated Annealing (SA)	Good approximation solution for a large size solution search space	No guarantee for the global optimum; highly expensive cost functions for problems with few local optima with smooth energy landscape	[21, 64, 112, 201, 203, 233]
Tabu Search (TS)	Efficient and effective in solution space searching; capable of incorporating multiple objectives	Time-consuming process	[81, 82, 83, 199, 215]
Variable Neighborhood Search (VNS)	Provides efficient and good approximation solutions although it is simple and does not require much change for extensions	Long extra computations may occur for highly constrained problems with many constraints to check and penalties to compute	[93, 146]

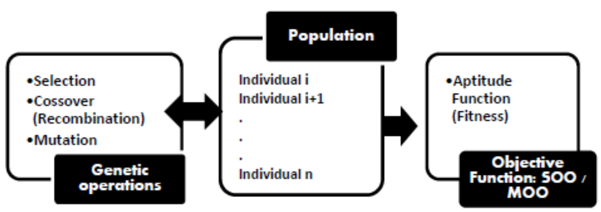


Fig. 2. The structure of general GAs.

II summaries the metaheuristics along with an analysis of their pros and cons.

1) Evolutionary Algorithms (EAs): EAs are characterized by metaheuristics, which refer to high-level algorithmic strategies. *Metaheuristics* guide other heuristic algorithms while searching through the feasible solution space for identifying an optimal solution. The underlying idea is to quickly converge toward an optimal solution [84]. Many NP-complete problems (e.g., task assignment / scheduling, team formation, traveling salesman problem) have been solved using this technique. This technique has the following characteristics:

- Identified feasible solutions tend to converge toward optimal or near-optimal solution(s);
- If optimal solution(s) are not identified, it may provide a good approximation with a small constant factor (e.g., 5%) of optimal solutions in many cases;
- It can perform even better under a special condition that non-zero probability of occurrence of each solution is guaranteed. Under this condition, an optimal solution can be found even if the set of solutions is non-convex;
- It significantly increases chances to identify global optima by avoiding local optima.

EAs have been used to solve "combinatorial optimization problems" whose optimal solutions can be obtained only with a very high computational cost. A common type of EA is the genetic algorithm (GA). In this context, evolutionary processes involve three mechanisms: *recombination* (or crossover), *variation* (mutation) and *natural selection*.

Recombination involves the creation of an offspring from previous entities (or an entity). During recombination, mutations may occur. As a result of these changes, the fittest entities survive while the weak ones die and do not produce offspring. To mimic these mechanisms, GAs use a general form of an evolutionary structure to solve these problems, as shown in Figure 2.

We briefly describe how GAs can be used to solve MOO problems. As shown in Figure 2, we assume that 'an individual in a population of size n' indicates a 'solution' to a MOO problem. One can run the GA on a population to identify an optimal solution to a particular problem. Given a set of genetic operations (selection, crossover, and mutation), new offspring or individuals are created from existing individuals either through mutation or crossover [64, 84, 168]. By iterating the genetic operators through the population, an optimal solution may be identified. The individuals or solutions are evaluated according to the aptitude function.

The final optimal solution values are derived from the processes of optimization and decision making. In Figure 3, we explain the decision process based on the three stages [98, 205].

Pros and Cons: EAs are devised to provide heuristic

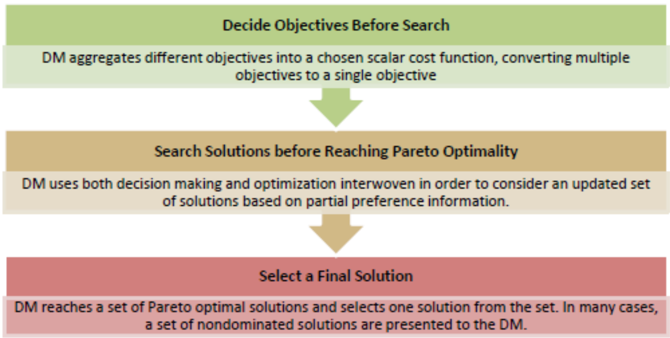


Fig. 3. Three key stages of the decision process in EAs [205].

methods capable of generating useful solutions for solving optimization problems. However, DMs often face computationally expensive fitness functions. Besides the solutions may converge towards local optima [205].

EAs that are designed to solve multi-objective optimization problems are called multi-objective evolutionary algorithms (MOEAs). The fundamental principle of MOEAs is approximating the solution to the Pareto-front and preserving the diversity of populations so as to obtain more nondominated solutions. Below we discuss several MOEA algorithms used to solve MOO problems in the literature.

a) Nondominated Sorting Genetic Algorithm (NSGA): NSGA aims to improve the adaptiveness of a population of candidate solutions based on a Pareto front constrained by a set of objective functions. The population is ordered based on a hierarchy of sub-populations according to the degree of Pareto dominance. Similarity of members in each sub-group is used to evaluate the Pareto front solutions to identify a diverse front of non-dominated solutions [55].

Srinivas and Deb [192] describe a two-step NSGA: *fast nondominated sorting* and *crowding distance calculation*. *Fast nondominated sorting* emphasizes elitism to preserve the best individuals until the current generation while *crowding distance calculation* promotes solution diversity.

In each generation, a parent population is sorted according to non-dominance using tournament selection. Fast nondominated sorting ranks all population (parents and children) into different levels of fronts, assigning a corresponding fitness value. Inside each front, a crowding comparison is conducted, representing the distance between neighboring solutions. With these two values (i.e., fitness value and crowding distance), if there are two solutions with different non-domination ranks, the lower rank will be preferred. If both solutions belong to the same front, a point in a less dense region (i.e., high crowing distance) will be preferred. These are to maintain solution diversity [55]. NSGA-II is later proposed to improve NSGA to further conserve elites and sort them in a fast procedure. NSGA-II is nonparametric, dealing with different noise distributions in objective functions to handle uncertainty [55]. It does not require particular characteristic structures, or parameters of data or population, using ranks rather than actual values [56].

Pros and Cons: NSGA preserves the explicit diversity of Pareto-optimal solutions by using a crowding distance comparison process. The elitism with nondominated sorting conserves already found Pareto optimal solutions [47]. These two features contribute to computational efficiency. However, poor performance may happen when a crowding distance function fails to generate the crowd of solutions under multiple objectives [120].

b) Strength Pareto Evolutionary Algorithm (SPEA): SPEA starts with an initial population and an empty archive. First, all nondominated population members are saved in the archive. During this update operation, any dominated individuals / duplicates are dropped from the archive. For example, if the size of the updated archive is larger than a predefined limit, archive members selected by a clustering technique preserving the characteristics of the nondominated front can be deleted. After the update, both archived and population members are given fitness values (smaller is better for minimization). When selecting a mate, binary tournaments are used to select individuals from the combined pool of population members and the members in the archive. Each individual in the archive is more likely to be selected than any population member. After recombination and mutation processes, the old population is updated with the new offspring population. SPEA2 is devised to improve issues of SPEA in assigning fitness values, estimating density based on a clustering technique, and truncating individuals from the archive. In particular, SPEA2 provides better solutions when the number of objectives increases. Unlike other MOEAs, SPEA2 aims at distributing solutions uniformly near the Pareto-optimal front [234].

Pros and Cons: Elitism (the archive) is used to preserve the previous best solutions and thus it has a better search ability. However, density estimation dominates the run time of the fitness assignment procedure [187].

c) Pareto Archived Evolution Strategy (PAES): PAES uses a simple evolution strategy, selecting a better solution out of two individuals where an individual solution is generated as a mutant. Only a solution that is better than the current solution is selected; otherwise, the solution is not updated. That is, it uses a local search to find diverse nondominated solutions [114].

Pros and Cons: PAES is the simplest possible non-trivial MOEA solver aiming to obtain Pareto optimality and solution diversity. The comparison between individuals using the simple evolutionary strategy contributes to reducing computational complexity. However, it may generate disconnected Pareto fronts [47].

d) Multi-objective Quantum-inspired Evolutionary Algorithm (MQEA): MQEA is derived from a quantum evolutionary algorithm (QEA) to solve multi-objective problems [111]. QEA employs a new probabilistic representation, called “Q-bit,” by using the concept of “qubits.” A qubit can be in a binary state such as 0 or 1 or in any superposition[1] of the two. A Q-bit individual refers to “a string of Q-bits”[2] and the length of Q-bit means the population size. Since the Q-bit

[1]When more than one quantum states are combined, we call it “superposed” [111].

[2]A Q-bits is the smallest unit of information in QEA [111].

can be probabilistically presented as "a linear superposition of states," it is useful for generating solution diversity in the evolutionary process [111].

In each generation, binary solution sets are formed when Q-bit individuals are observed multiple times from the previous (or last) generation, and then the fitness level of every binary solution is evaluated. A rotation is applied to each individual to update the state for producing the next generation. The best individuals are preserved from a set of sorted nondominated solutions among current solutions and previous best solutions, i.e., ranking each in terms of nondominated front, and the global optimal solution pool is updated.

Pros and Cons: MQEA takes advantage of quantum computing for mutation and crossover procedure, which is scalable. It can improve proximity to the Pareto optimal front and diversity of nondominated solutions. However, it may be computationally intensive to find sufficient convergence conditions to Pareto optimal front [126].

e) Hierarchical Evolutionary Algorithm (HEA): Instead of exploring nondominated individuals in parallel, HEA tackles each objective in sequence [61]. In each epoch, it starts with one of the objectives, finding the best individuals by applying mutation evaluated by the fitness value; afterwards it feeds those solutions into the next objective as input and follows the same way as before to explore the best solution out of the feasible solutions that met the previous objective [61].

Pros and Cons: HEA is computationally efficient by reducing the complexity in ranking each individual by screening out some invalid or dominating solutions for each objective. However, if all the solutions in the search space are feasible (i.e., none filtered out), then there is no reduction in computational complexity [61].

2) Ant Colony Optimization (ACO): It is inspired by observing the foraging behavior of certain ant species. Ants cooperate with each other to seek food efficiently via indirect communication [24, 63]. First, scouting ants begin searching for sources of food. When the ants return to the colony, they leave pheromones, as markers on the path leading to the food, which will give other ants directions to find the food [24, 63]. More pheromones on a path mean a better path with higher probability of finding food.

Ants find solutions by traveling on the graph, and accordingly pheromones on each path can be updated. Through the indirect communications, ants can use the pheromone value in choosing a path. This ultimately leads to finding a better (or best) path. To use this method, an optimization problem should be converted to a problem of finding a best path on a weighted graph. Dridi et al. [65] apply a bi-colony ACO to solve resource assignment and task scheduling problems for surveillance systems.

Pros and Cons: ACO is useful for dynamic applications such as finding routes under dynamically changing network topologies. Cooperative behaviors of entities in distributed systems can lead to rapid discovery of good solutions and also avoid premature convergence to the final solution. However, even if convergence is guaranteed, there may be uncertainty in the time taken to convergence [24, 63].

3) Particle Swarm Optimization (PSO): This technique is inspired by the social behavior of flocking birds or schooling fish [109]. Simple entities, also called the particles, are formulated on the parameter space of a function (or problem), and each calibrates the fitness at its current state (i.e., location). Then each particle makes a decision for its next movement through the parameter space by combining its own previous fitness values and additional members of the swarm. After that, the particle can be evaluated by the states and fitness values of other particles [109]. A social neighborhood is formed based on the members of the swarm with which a particle can interact. Jin et al. [104] apply hybrid PSO algorithms to solve task allocation in wireless sensor networks.

Pros and Cons: PSO is a simple process of coding, independent of a set of initial points, and has fewer parameters than other heuristic optimization algorithms. Hence, it is capable of producing high quality solutions in much less time with more stable convergence behavior, compared to other stochastic methods. On the other hand, similar to other heuristic optimization techniques, since it does not have a mathematical basis for analysis, it is difficult to identify the optimal parameters in some cases [123].

4) Simulated Annealing (SA): SA is a probabilistic technique for identifying the global minimum of a cost function that may have a number of local minima [112, 201]. This technique emulates the process of cooling a solid slowly, which makes its structure ultimately 'frozen' at a configuration with minimum energy [21].

The underlying idea of SA is to obtain the optimal solution by decreasing the probability of accepting worse solutions slowly. This explores the solution space [112, 201]. Zhu et al. [233] use SA to solve a resource allocation problem in real-time distributed systems. Dorn et al. [64] employ SA to propose a team formation algorithm in online social networks.

Pros and Cons: SA is useful when the search space is large. For some problems requiring an acceptable level of optimization or restricting search time, SA is often more efficient than exhaustive search [112, 201]. However, SA is not always efficient for all combinatorial optimization problems. For problems with few local optima and a smooth energy landscape, the computational cost is too expensive. If the cost function in SA has narrow and steep valleys, it may not work efficiently. As is the case with other GA approaches and other bio-inspired approaches (e.g., ACO or PSO), SA also does not guarantee the global optimum [203].

5) Tabu Search (TS): TS is a metaheuristic search method using a local search method for mathematical optimization [81, 82, 83]. TS is based on two aspects of intelligence: "adaptive memory" and "responsive exploration".

TS searches local neighborhood of elite solutions (i.e., best or better solutions of the solution pool) hoping to find an improved solution. Local searches tend to get stuck in suboptimal regions or on plateaus with many equally acceptable solutions. To avoid this problem, TS uses adaptive memory structure to improve the performance of these techniques where the memory structure contains the visited solutions or sets of rules. By marking solutions previously visited within a certain short-term period or violated based on the user's rule as 'tabu' (forbidden), the algorithm does not repeatedly consider them.

TS also conducts *responsive exploration* by changing the solution space based on good or bad solution features. The memory structure in TS uses four dimensions: recency, frequency, quality, and influence [81, 82]. The key benefit of using the adaptive memory structure in TS is the balance between search intensification and diversification by delving into attractive regions more thoroughly while expanding search spaces which have not been explored previously [81, 82]. For more details on TS, readers are referred to [81, 82, 83]. Weerakoon and Allan [215] apply TS to form coalitions for optimization of secure routing in wireless sensor networks.

Pros and Cons: Since TS adaptively remembers the 'tabu' solutions, it searches the solution space more efficiently and effectively, compared with memoryless approaches, such as many GA and SA metaheuristic methods. In addition, TS is capable of incorporating multiple objectives, overcoming the inherent limitations of GA and SA methods. However, TS requires a large number of evaluations for every optimization step, consuming significant time and effort, although the computational efficiency increases by finding feasible solutions [199].

6) Variable Neighborhood Search (VNS): It has been proposed to solve combinatorial optimization problems based on the distance between a current solution and and its neighbors, representing local optima, leading to a new improved solution [146]. VNS uses the idea that neighborhoods change both in descent to local optima and in escape from the valleys containing the local optima [93]. VNS employs the following facts to find optimal solutions: (1) local optima for one neighborhood structure are not necessarily local optima for another neighborhood structure; (2) a global optimum is a local optimum based on all possible neighborhood structure; and (3) local optima for one or more neighborhood structures are relatively close to each other in most problems [93]. The common variants of VNS include: (1) variable neighborhood descent (VND) based on a purely deterministic search; (2) reduced VNS (RVNS) using stochastic search; (3) skewed VNS (SVNS) providing a way to escape from very large valleys; and (4) variable neighborhood decomposition search (VNDS) using two-level VNS. For more details on VNS, interested readers are referred to [93, 146].

Pros and Cons. Extensions of VCN are relatively simple and require no changes of parameters and provide quite good approximation solutions. However, there is additional time complexity of computing the penalty and checking multiple constraints at every move and swap.

Overall metaheuristics algorithms provide useful methods to search a large solution space to solve NP-complete or NP-hard MOO problems. Obviously, they cannot guarantee global optimal solutions.

C. Hybrid Metaheuristics

The motivation for combining multiple methods derives from the intention of uniting benefits of individual approaches [166]. The hybridization is not limited to just merging two methods, but can include combining metaheuristics with exact algorithms [26]. In this section, we discuss the hybrid approaches that particularly combine metaheuristics with other approaches, either exact algorithms generating optimal solutions or other metaheuristics producing approximation solutions. Table III summarizes hybrid metaheuristics methods with an analysis of pros and cons.

1) Ant Colony Optimization (ACO) + Constraint Programming (CP): We omit the description of ACO as we discussed the ACO in Section IV-B2. In CP, a filtering algorithm, associated with constraints, aims to delete values from variable domains that are not contributing to feasible solutions. CP has two phases: (1) the propagation phase to remove values not in feasible solutions by the filtering algorithm; and (2) the labeling phase to an unassigned variable that is assigned with a value from its domain where only improving solutions are considered as feasible solutions. By combining ACO and CP, the advantages of both methods can be united based on ACO's learning capability and CP's ability to handle constraints efficiently [26].

Pros and Cons. Compared to a pure ACO, combing with CP can provide more efficient but good quality solutions [26]. However, whenever a new, improved solution is found, a global constraint may need to be updated to enforce the improvement of all constructed solutions, which incurs extra overhead.

2) Variable Neighborhood Search (VNS) + plus Large Neighborhood Search (LNS): The critical concern of using VNS is the choice of a neighborhood structure that generates neighboring, feasible solutions [26]. The common tradeoff between the size of neighboring solutions and efficiency is shown in that a small set of neighboring solutions generates efficient but low quality solutions while a large set of neighboring solutions produces slow but high quality solutions. Finding the optimal solutions with the neighborhood is actually a NP-hard problem. Nowadays the research is known as *Large Neighborhood Search* (LNS) which generates efficient and high quality solutions. LNS can be efficiently explored by complete methods (i.e., methods guaranteeing optimal solutions) such as constraint programming, mixed-integer programming, or dynamic programming [162, 164, 26].

Pros and Cons. LNS can complement VNS by finding good quality neighborhood solutions efficiently. However, finding LNS using complete methods is also a NP-hard problem [26].

3) Tabu Search (TS) + Problem Relaxation (PR): Relaxation techniques are used to relax constraints for difficult combinatorial optimization problems by simplifying certain constraints using branch and bound algorithms, dropping integrality constraints, or moving constraints to enhance a particular objective [26]. The information gained from relaxations can allow a greedy manner of generating solutions [26]. The hybrid approach combining TS with the problem relaxation technique has two steps: (1) relax constraints by dropping the integrality constraints; and (2) apply TS to search for optimal solutions on the relaxed problem. An example of solving a multidimensional 0-1 knapsack problem using this hybrid approach is shown in [204].

Pros and Cons. Problem relaxation allows efficiently finding feasible solutions by relaxing constraints in various ways.

TABLE III
HYBRID METAHEURISTICS METHODS

Technique	Pros	Cons	Ref.
ACO + CP	Generates efficient but good quality solutions by leveraging the benefit of using CP	Update of global constraints incurs extra overhead	[26]
VNS + LNS	Provides good quality neighborhood search region	Finding LNS using exact algorithms is NP-hard	[26, 162, 164]
TS + Problem Relaxation	Finds efficient feasible solutions with problem relaxation	May not provide optimal solutions	[26, 204]
SA + TS	Controls worsening solutions using SA's temperature parameter	Computationally expensive for problems with few local optima	[13]
EA + DP	Produces efficient feasible solutions	May reduce solution diversity	[25, 26]

However, due to the relaxation, the hybrid approach tends to provide approximation solutions rather than optimal solutions [26].

4) Simulated Annealing (SA) + Tabu Search (TS): TS performs movements to update solutions with improvements. However, when no better movements are available, TS goes into a second phase by moving to a worse solution, hoping that a better solution can be found. In the hybrid approach of SA and TS, when TS reaches the state that does not find better solutions, it lets SA control the worsening movements by the temperature parameter [13].

Pros and Cons. This hybrid approach can take advantage of the benefits of both metaheuristics in that SA is used in the second phase of TS to more systematically adjust the parameter to control worsening solutions which can increase the odds of finding improved solutions. However, still whether to accept worsening solutions or not should be analyzed. Further, for a solution space with a smooth energy landscape with few local optima, the computational cost of finding the solution is high.

5) Evolutionary Algorithm (EA) + Dynamic Programming (DP): DP provides a way to define optimal strategies that can be used in various metaheuristics [26]. Blum [25] shows a hybrid approach showing how DP is used in evolutionary algorithms within the crossover operator that generates candidate solutions with improvements based on the extension of previous solutions.

Pros and Cons. EA can have the benefit of efficiency introduced by the speed-memory tradeoff in DP in generating candidate solutions to make improvements in feasible solutions. However, as DP is basically extending a solution based on partial solutions from previous solutions, it may be stuck with local optima due to less solution diversity [26].

In addition to the above hybrid metaheuristics approaches, Mezmaz et al. [137] combine two metaheuristics, GA and a memetic algorithm, with an exact algorithm, a branch and bound (BB) algorithm to guarantee optimality. Bedoui et al. [19] also present the combination of NSGA-II and TS and the combination of NSGA-II and SA to solve MOO problems. For most hybrid metaheuristics, exact algorithms or other metaheuristic approaches are utilized to improve solution efficiency which is often traded off for solution optimality. However, the main benefit of using hybrid metaheirstics is better efficiency for the same or improved solution optimality.

D. Trust-based Algorithms

Trust-based MOO problem solutions have received significant attention as a promising heuristic to solve MOO efficiently and effectively under resource-constrained, hostile, distributed network environments [36, 50, 64, 84, 181]. Although trust-based solutions may not perfectly generate Pareto optimal solutions, it has been used to generate approximation solutions that are close-to-optimal with significantly high efficiency (e.g., exponential to linear) [40, 213].

The definition of trust is "assured reliance on the character, ability, strength, or truth of someone or something" [136]. In essence, trust is a relationship in which an entity called the trustor relies on someone or something called the trustee, based on given criteria. As trust is a multidisciplinary concept, the term has been used in different disciplines to model different types of relationships: trust between individuals in social or e-commerce settings, trust between a person and an intelligent agent in autonomous systems and trust between network entities in communication networks [41]. The nature of trust is subjective and domain specific. Thus, the concept of trust is often defined by specific criteria and the application context.

For example, trust-based MOO approaches have used metrics such as feedback credibility, service satisfaction, preference similarity [50], a team member's qualification [64], network performance (e.g., security, reliability, load balance, or throughput) [181], and so forth. The basic idea of trust-based solutions is to dynamically assess trust status of all nodes in the system, and then factor this knowledge into the MOO problem formulation. Very often trust-based heuristics can be used to reduce the solution complexity at the expense of solution quality.

Pros and Cons: Trust-based solutions are heuristic in nature and may produce close-to-optimal solutions with linear complexity as good approximation methods. This efficiency can increase its applicability in resource-constrained environments as one of key concerns is to minimize energy consumption in solution search. However, the heuristic method is implemented in a greedy manner, which does not guarantee optimal solutions.

V. GAME THEORY-BASED MOO DESIGN TECHNIQUES

Game theory has been used to solve MOO problems as design tools [35, 169, 189]. Game theory is similar to MOO

TABLE IV
GAME THEORETIC APPROACHES

Approach	Technique	Pros	Cons	Ref.
Cooperative Game Theory	Nontransferable Utility Game (NTU)	Generic; nonparametric	No guarantee of a unique solution unless a certain condition is met	[29, 152, 155, 176, 190]
	Hedonic Game	Generic; high predictability	Requires additional conditions to ensure stable partitioning for different presentations	[10, 12, 23, 28, 176]
	Shapley Value	Simple to measure utility for a coalition	High communication overhead; no guarantee for obtaining information under hostile environments	[77, 143, 179]
	Core	Provides a set of stable coalitions	Hard to select a best coalition when the size of the core is large	[71, 174]
	Nash Bargaining Solution (NBS)	Generic with less complexity	Not straightforward for a cooperative concept; non-trivial to evaluate axioms	[152, 194]
Auction Theory	Vickrey's Auction	Provides a dominant strategy in decision making	No bidding may occur if prior knowledge on the real valuation of an item is not known; vulnerable to the auctioneer manipulating prices	[9, 27, 183]
	Vickrey-Clarke-Groves (VCG) Auction	Effective if players play with truth-telling as a dominant strategy	May not achieve budget balance; cannot track bidder's real identities; vulnerable to collusion attacks	[9, 27, 89, 118, 152, 160]
	Combinatorial Auction	Incurs less amount of communications; reveals less private information	Introduces complexity in selecting items for each bundle	[27, 42, 68, 152]
	Reverse Auction	Maximizes the social welfare; encourages the competitiveness of sellers	Vulnerable to seller collusion attacks; relies on whether buyers play truth-telling	[68, 152]

in that both aim to optimize multiple objectives at the same time. However, in game theory, each player's utility function is controlled by a different agent where it is the part of decision variables to optimize the performance. In addition, it is a well-known fact that the goal of MOO is to find Pareto optimality while non-cooperative game theory aims to identify Nash Equilibrium (NE) which may not be PO, as shown in the Prisoner's Dilemma [169]. We discuss two game theoretic approaches (i.e., cooperative game theory and auction theory) in this section as useful design tools for a system with multiple objectives based on the following reasons: (1) in cooperative game theory, an identified NE solution may be Pareto optimal because individual players are cooperative to achieve a common goal of maximizing the payoff of a coalition; and (2) in auction theory, incentive-compatibility design can stimulate individual players to maximize a system goal, leading to an NE equilibrium (if it exists) which is Pareto optimal. Table IV summaries the two game theoretical approaches and the associated solution techniques for MOO problem solving, along with an analysis of pros and cons.

A. Cooperative Game Theory

Many MOO problems, such as coalition formation (or team composition), task assignment, task scheduling, or resource allocation, are formulated as an *n-person* cooperative (coalitional) game.

A cooperative game appears when groups of players, called coalitions, cooperate to obtain benefits by joining a grand coalition, the set including all coalitions. The game is played by coalitions of players, not by players within each coalition. A cooperative game is also called a "coalitional game" [29, 152]. Often strategic games assume that individual players are selfish and maximize their utility, assuming no cooperation. The goal of a cooperative game is to model the situations in which the players work together or share some cost to benefit each other. However, players are selfish in that they are cooperative only if their cooperative behavior maximizes their utility. Cooperative game theory aims to understand mechanisms for players to cooperate in order to maximize the payoff of a grand coalition [29, 152].

A cooperative game has two key attributes: (1) a set of players $N = \{1, 2, \dots, n\}$; and (2) a "characteristic function" v that computes the value obtained from subsets of N: $v(S)$ is the value associated with a coalition formed with all the members in S. A cooperative game can be denoted as a pair (N, v) [29].

The objective(s) of the system must be reflected in function v and should be achieved in coalition payoff. An individual player can calculate its payoff based on its objective while a coalition leader can compute the coalition payoff given a set of selected members. Incentive (or reward) or penalty mechanisms can be utilized to enforce desired behaviors of individual entities that can contribute to increasing coalition payoff. Many existing works use various types of cooperative games to solve coalition or team formation problems such as nontransferable utility cooperative games [190], hedonic games [176], and repeated cooperative games [97]. Some existing works use trust in formulating the payoff functions where trust can be the basis of decisions by players [29, 84, 87, 97, 141, 150].

Pros and Cons: Although cooperative game theory provides a generic framework for solving coalition problems in many domains, the assumptions or rationale under this theory may not always be true. For example, even though the commitment of a player can be ensured for honest players, when attackers exist in cyberspace, the rationale of the cooperative game no longer holds. Hence, some variation of this game is needed to consider attacker behaviors. In addition, as a common issue in using game theory, the assumption of a player's rationality (i.e., be rational to maximize its utility) may not always be guaranteed, particularly when human entities are involved and could be affected by their propensity or emotions in the decision making process. Also humans may not have the computational capacity, given deadlines, to compute optimal strategies. That is why there is a significant amount of research on approximating optimal strategies. Below we briefly discuss some examples of cooperative game theories.

Now we discuss two common game theoretic approaches used to formulate cooperative games, *Nontransferable Utility Game* and *hedonic game*. In addition, we describe solution concepts of cooperative games including *Shapley value*, *Core*, and *Nash Bargaining Solution*.

1) Nontransferable Utility Game: Cooperative games are generally referred to as non-transferable utility (NTU) games. A NTU game is denoted as a pair $(\mathcal{N}, \mathcal{V})$, where $\mathcal{N}$ is a set of players and $\mathcal{V}$ is a coalitional function assigning payoff to each coalition. $\mathcal{S}$ is a subset of $\mathcal{N}$, $\mathcal{S} \subset \mathcal{N}$ and the feasible payoff vector of $\mathcal{S}$ (i.e., payoff coalition S can generate) is denoted by $\mathcal{V}(\mathcal{S}) \subset \mathbb{R}^{|\mathcal{S}|}$. The feasible payoff vector, $\mathcal{V}(\mathcal{S})$, is nonempty for $\mathcal{S} \neq \emptyset$, "closed, convex, and comprehensive" (i.e., $x' \leq x \in \mathcal{V}(\mathcal{S}) \Rightarrow x' \in \mathcal{V}(\mathcal{S})$) [155]. *Nontransferable utility* means that agents do not have a common scale to measure the value of a coalition. Likewise, NTU can be defined by a tuple $(\mathcal{N}, \mathcal{X}, \mathcal{V}, (\succ_i)_{i \in \mathcal{N}})$ where $\mathcal{X}$ is a set of outcomes/coalitions, $\mathcal{V}$ is an outcome function for a coalition, and $\succ_i$ indicates the relation of agent i's preference over the set of coalitions. The relation is assumed to be transitive and complete. Singh et al. [190] model a resource allocation problem in commercial wireless networks as a nontransferable payoff coalitional game.

Pros and Cons: NTU is generic as it can be used to represent many other cooperative games, i.e., hedonic games. Since it uses preferences instead of utilities (i.e., prices and quantities), it does not impose any functional form on the utility function so it is nonparametric [39]. In addition, when trust between agents is utilized as a factor in their coalition formation, we can derive preferences among the agents directly from their trust values. However, NTU does not guarantee a unique solution to the game unless certain conditions are satisfied.

2) Hedonic Game: A hedonic game is a special NTU and can be defined by a tuple $(\mathcal{N}, (\succeq_i)_{i \in \mathcal{N}})$ where $\mathcal{N}$ is a finite set of players and $\succeq_i \subseteq 2^{\mathcal{N}_i} \times 2^{\mathcal{N}_i}$ is a "complete, reflexive and transitive relation" for agent i's preference, implying that if $\mathcal{S} \succeq_i \mathcal{T}$, agent i prefers coalition $\mathcal{T}$ with the maximum of coalition $\mathcal{S}$ [12, 28]. No externalities are considered across groups. That is, players' preferences can be only expressed over the set of coalitions to which players belong [23]. Cooperative games can be viewed as hedonic games if the payoff to a member in a coalition is only affected by the coalition members and can be predicted, such as voting. Saad et al. [176] model a task allocation problem as a "hedonic coalition game" in multi-agent wireless networks.

Pros and Cons: Like NTU, hedonic game theory is also generic and can be applied to various problems without much complexity. Compared with other cooperative games, hedonic games can give high predictability by restricting a player's preferences only to the coalitions to which it belongs. However, in order to ensure stable partitioning for different representations, hedonic games may require additional conditions. This remains an open research problem [10].

3) Shapley Value: The Shapley value indicates how important each player is to the overall cooperation and what payoff each player can expect from the coalition based on the contribution provided by the player. The Shapley value is computed as:

$$\varphi_i(v) = \sum_{S \subseteq N \setminus i} \frac{|S|!(n - |S| - 1)}{n!} \left(v(S \cup \{i\}) - v(S)\right) \quad (15)$$

The Shapley value indicates the amount of profit player i obtains in a given coalition game. $N \setminus i$ is a set of players that does not include player i. S is a subset of N and $v(S)$ is the value of the coalition S, the total expected gain from coalition S. $v(S \cup i - v(S))$ gives the fair amount of player i's contribution to the coalition game. This is averaged over the possible different permutations where the coalition can be formed.

Garg et al. [77] and Militano et al. [143] use the Shapley value to identify an optimal coalition that maximizes multiple objectives associated with players and coalition leaders.

Pros and Cons: The Shapley value gives a simple way to measure the utility a coalition can obtain through a particular entity. However, in order for opinions/status of other entities to be disseminated over the network, it incurs high communication overhead and there is no guarantee for the information to be delivered without any change in the presence of malicious entities. Thus, the Shapley value may not be easily implementable in dynamic, distributed adversarial environments.

4) Core: It represents a profile of payment divisions, which guarantees no player has incentive to leave a current coalition to form another coalition $\mathcal{S} \subset \mathcal{N}$ [174]. "Rational" players are assumed to agree to the grand coalition because no other coalition can give better payoff. The main problem is how to divide the payoff so that all participating coalitions are satisfied with the assigned payoff [71]. The *core* in a transferable utility (TU) game, $\mathcal{C}_{TU}$, is defined as a set of vectors $\mathbf{x}$ in which x_i's refer to distributions to payoff that are efficient and considered as rational by individuals. $\mathcal{C}_{TU}$ is given by

$$\begin{aligned} \mathcal{C}_{TU} = \{x \in \mathbb{R}^{|\mathcal{N}|} : &\sum_{i \in \mathcal{N}} x_i = v(\mathcal{N}), \\ &\sum_{i \in \mathcal{S}} x_i \geq v(\mathcal{S}), \ \ \forall\, \mathcal{S} \subseteq \mathcal{N}\} \end{aligned} \quad (16)$$

where the first condition indicates *efficiency* that the sum of payoffs to members is the payoff to the grand coalition. The second condition called *coalitional rationality* is that the grand coalition's payoff is no less than any participating coalition's payoff.

Pros and Cons: The core addresses the issue of incentives for abandoning a grand coalition, under which it provides a set of stable collaboration that no player will leave to form another coalition. However, the cores (of TU or NTU games) are not always guaranteed to exist. In zero-sum games, the core is empty because no coalition is satisfied with a negative payoff, thus exhibiting strong instability. On the other hand, the size of the core can be large where there is high stability (i.e., highly balanced payoffs to make all coalitions satisfied), but selecting a suitable coalition is challenging due to its high complexity [174].

5) Nash Bargaining Solution (NBS): In cooperative games, NBS can be applied in a situation in which two or more players are required to select one of possible outcomes derived from a joint collaboration [152]. Particularly when two parties negotiate something associated with each party's interest, a bargaining game may result in a disagreement outcome, i.e., a payoff each player receives when a negotiation is not successful. Examples are when an employer negotiates with a potential employee on salary or when two countries negotiate on trade issues. We refer readers to [152] for details of NBS.

Nash's bargaining theorem uses the concept of a collective utility function that can be maximized by the unique solution identified in the bargaining game [152]. The collective utility function refers to a function aggregating the utilities of individuals into a single value indicating the utility of the collective. Examples include the utilitarian function (adding all individual utilities up), the egalitarian function (taking the minimum of individual utilities), and the Nash function (taking the product of the utilities) [152]. Subrata et al. [194] model a job allocation problem in grid computing as a cooperative game and identified the task allocation structure based on NBS.

Pros and Cons: NBS provides a generic solution strategy for various cooperative bargaining processes without adding much complexity. However, it is not straightforward to select and adjust a cooperative concept to reflect unique characteristics of a given bargaining situation.

Note: Although cooperative game theory (CGT) is commonly used to solve diverse MOO problems, non-cooperative game theory (NCGT) is also used to solve MOO problems. The distinction between CGT and NCGT is that CGT has a group as a basic unit while NCGT has an individual as a basic unit [186]. A cooperative game aims to achieve a value (e.g., maximization of a grand coalition's payoff and each coalition's payoff) where there is a force from a third party (e.g., coalition leaders) to cooperate while non-cooperative game does not have any such forcing function, but operates based on an individual rational player's self-interest. Thus, even if NCGT is used, cooperation may occur but the motivation of the player's action is only to maximize its own utility. That is, in NCGT, a player takes an action only based on its self-interest [186].

A few existing works use NCGT to mainly solve resource allocation problems with multiple objectives [34, 90, 207, 226]. However, since in NCGT a player makes a decision only based on local view, using NCGT for solving MOO problems often results in sub-optimal solutions while incurring less overhead than in CGT [191].

B. Auction Theory

An auction can be used to make a deal between willing sellers (auctioneers) with items to sell and buyers (bidders) willing to pay a price to purchase them [118]. Auction theory is particularly suitable for modeling a coalition formation MOO problem in which coalition leaders can be auctioneers, while members can take the role of bidders. In the coalition formation problem, coalition leaders are attempting to maximize their payoff by recruiting the best members while the potential bidders want to maximize their gain by attaching to coalitions with high payoff. So, the bidders will bid and commit to buying items (joining coalitions) where the bid provides the best individual payoff. In this way, the coalition payoff (i.e., the auctioneer's criteria to determine winners) will influence the member selection process and thus the coalition formation (or team composition). Figure 4 shows the decision process and objectives of the auctioneers and the bidders.

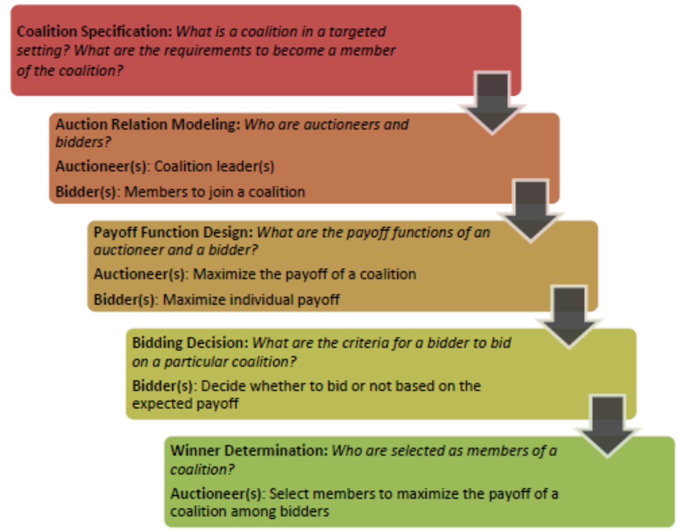


Fig. 4. Key components of auction-based game for coalition formation problems.

Given multiple items, a potential bidder can have different degrees of preference for each item. Thus, the payoff towards an item may be different depending on the bidder's value and needs. The payoff bidder j can obtain from purchasing item i is the net gain of the valuation $v_{i,j}$ at the expense of the price p_i for item i, $v_{i,j} - p_i$ [27].

Even if bidder j bids on item i, bidder j is not a final winner of item i unless the auctioneer agrees with it. The auctioneer should select a bidder that can best maximize the payoff. The payoff that auctioneer k can obtain from accepting bidder j is the net gain of the valuation $v_{k,j}$ by selecting bidder j at the price of p_j, $v_{k,j} - p_j$ [27]. Based on this process, a set of coalitions is formed. However, when the contract is terminated, a new coalition reformation process can be triggered. Often a bidder's contract termination may cause loss to the bidder's payoff as the system penalizes the bidder's lack of commitment towards the initially assigned coalition. Note that a bidder can bid for multiple coalitions in parallel based on the characteristics of a coalition, e.g., a coalition executes a short-term or a long-term mission, which determines whether to free selected members after finishing the mission.

Coalition formation problems have been solved using many different types of auction-based algorithms such as Vickrey's second price auction [183], Vickrey-Clarke-Groves (VCG) mechanism [89], a single-item auction with multiple preferences [36], a combinatorial auction with multiple preferences [42], a reverse auction [68], Nash bargaining [194], and an auction based on different bidding strategies [217]. Here we briefly discuss the *pros* and *cons* of each auction-based algorithm.

1) Vickrey's Auction: In single-shot auctioning, an auctioneer chooses a bid with the highest bid price. On the other hand, a bidder's best strategy depends on what it knows or guesses about the strategies of other bidders. Deciding what value to bid is not a trivial problem. To resolve these bidding problems, Vickrey's auction mechanism, called "second price auction", is proposed as a "sealed-bid auction" [27]. In this mechanism, a winning bidder pays the second highest price. This policy makes truth-telling as the "dominant strategy," regardless of the strategies of all other bidders, to maximize a bidder's efficiency because the winner pays the second highest price, not the price the winner has bid. Shi et al. [183] use Vickrey's auction to form a coalition of secondary users in cognitive networks.

Pros and Cons: Vickrey's auction is efficient in decision making, and results in a "dominant strategy" if bidders provide a truthful value [9]. However, if a bidder does not know the real valuation of an item, it may not bid on it at the market price without sequential auctions. In addition, if an auctioneer manipulates pricing via shill bids (i.e., bidding on an item to artificially increase its price), the price a winner would pay may increase [27]. In this sense, a bidder's trust in an auctioneer will impact the bidder's decision on whether to bid on an item offered by a particular auctioneer.

2) Vickrey-Clarke-Groves (VCG) Auction: VCG auction is a sealed-bid auction for auctioning multiple items. Bidders have their valuations of the items. In this auction, an individual bidder is charged with the loss it introduced to other bidders [27]. This mechanism also ensures that a truth-telling bidding is a player's dominant strategy to maximize the so called "social welfare", the sum of all players' utilities [27]. Guo et al. [89] propose a variant of VCG to model secure key management in mobile networks.

Pros and Cons: VCG mechanism is effective where players play with truth-telling as a dominant strategy. However, for problems aiming to achieve social welfare, even if an individual player may reach efficiency, a budget balance may not be achieved. Thus, when the total social welfare is greater than the cost to purchase an item (e.g., building a bridge), an efficient outcome based on selected winning bidders can be chosen [118]. In addition, if the auctioneer is not sure about the bidders' real identities, then a single bidder can use two different names and can win at a total price of zero [9]. Further, it is vulnerable to collusion that even losing bidders may receive "profitable joint deviations" at a very low price [9]. Vickrey auction and VCG auction mechanisms have been used to obtain good system-wide solutions by applying the concept of the so called *mechanism design* [152, 160].

3) Combinatorial Auction: In this auction, multiple items are auctioned simultaneously; bidders can express their multiple preferences over available bundles where a bundle refers to a collection of items. Winners, those who win the bid to purchase a selected bundle, are selected based on the bid prices. Cho et al. [42] devise a combinatorial auction algorithm (CAA) aiming to achieve a dynamic multiple mission assignment in military tactical environments. Li and Sycara [124] solve a coalition formation problem in online markets using CAA. Besides CAA was used to solve multiple dynamic coalition (or task assignment) problems [27].

Pros and Cons: Blumrosen and Nisan [27] discuss two main advantages of CAA: (1) reducing the amount of auctioneer-bidders communications; and (2) revealing less private information per bundle. However, how to select multiple items as one bundle significantly affects solution complexity due to combinatorial problems.

4) Reverse Auction: Reverse auction is a type of procurement auction by which several sellers offer their items for bidding, and compete for the price which a buyer will accept. Nisan et al. [152] describe that in a reverse auction, the buyer wants to procure an item from the bidder with the lowest cost, which may lead to maximizing social welfare. Based on the rule of VCG, the winning bidder only pays the second lowest price to the bidder who bids the item with the lowest cost and pays nothing to others [152]. Edalat et al. [68] use reverse auctioning for task allocation in energy-constrained wireless sensor network environments.

Pros and Cons: Reverse auction is commonly used for procurement. It may save a buyer money while maximizing the social welfare based on the sum of utilities of all buyers and it can encourage competitiveness among sellers (suppliers). However, if sellers collude in price fixing, it may not bring much benefit to buyers. Again buyers need trust mechanisms to assess trustworthiness of auctioneers for their colluding possibility.

VI. MOO Problem Classification

In this section, we use a novel way to classify existing MOO modeling and solution techniques based on the types of multiple objectives. This allows us to analyze the most common modeling and/or solution techniques used for each class of MOO problems to best trade solution efficiency vs. solution quality.

Existing work on MOO can be categorized into three groups depending on whether the work deals with system objectives for global welfare and/or individual objectives for individual welfare. ***Class 1*** represents the case in which there are multiple system objectives, but no individual objectives. In ***Class 2***, every individual entity has the same objective function. In ***Class 3***, each entity has its own individual objective function.

We cover a wide range of applications including coalition formation (or team composition), task assignment, task scheduling, and resource allocation based on the publications since 2000. In Sections VI-A, VI-B and VI-C, we survey modeling and solution techniques for Class 1, Class 2, and Class 3 MOO problems, respectively. In Section VI-D, we

TABLE V
EXISTING WORKS IN CLASS 1 WITHOUT USING TRUST

Ref.	System Objective	Techniques	Problem	Domain
[1]	Maximize both the number of completed tasks and task efficiency	NSGA; SPEA	Coalition formation	Multi-robot systems
[11]	Minimize workload and cost; Maximize system reliability	Quantum-based evolutionary algorithm	Task assignment	Distributed Systems
[22]	Maximize income while minimizing disturbance to workload	Goal programming	Resource allocation	Health systems
[19]	Maximize QoS to users; minimize hardware investment	Hybrid metheuristics combining NSGA-II with either TS or SA	Resource allocation	Broadcasting system
[38]	Maximize coalition utility while minimizing the number of robots involved in a task	Heuristic leader-follower based coalition algorithm	Task allocation	Multi-robot systems
[60]	Identify a path minimizing cost, time, and distance in directed and undirected networks	Hybrid metaheuristics with GA and VNS	Optimal path identification	Road networks
[61]	Minimize data volume and energy consumption; maximize quality-of-service	Evolutionary algorithm	Task allocation	Visual sensor networks
[65]	Minimize the time spent for project completion and associated cost	Bi-colony ant based approach	Task allocation	Mobile / stationary surveillance systems
[88]	Minimize task execution time while minimizing data transfer time	Particle swarm optimization	Task assignment	Cloud computing
[101]	Maximize throughput while minimizing resource consumption	ε-constraints	Resource allocation	Multi-hop wireless networks
[104]	Maximize task completion ratio while minimizing energy consumption	Genetic algorithm	Task allocation	Wireless sensor networks
[91]	Minimize task execution time and resource consumption	Fuzzy logic, genetic algorithm	Task assignment	Manufacturing systems
[134]	Maximize speediness of task execution and assignments functionality; minimize risk due to allocation decision	ε-constraints	Task allocation	Multi-agent systems
[137]	Find a schedule of a set of jobs on a set of machines that minimizes the total completion time and the total tardiness	Hybrid metaheuristics combining GA and memetic algorithm with a branch and bound algorithm	Flowshop scheduling problem	Manufacturing production systems
[142]	Maximize social benefit; minimize loading conditions to maintain voltage stability	Market-based pricing mechanism	Resource management	Decentralized electricity markets
[154]	Maximize task execution quality; minimize energy and bandwidth consumption	Genetic algorithm	Task assignment	Grid computing
[158]	Maximize throughput; minimize energy consumption	Linear programming	Resource allocation	Wireless mesh networks
[165]	Maximize security and QoS; minimize energy consumption	NSGA-II	Resource allocation	Wireless sensor networks
[182]	Maximize profit margins and the total importance of selected business partners	Fuzzy logic; entropy	Resource allocation	Manufacturing systems
[188]	Maximize the parallel optimization of task assignment and resource allocation	Generalized particle model	Task allocation	P2P and grid computing
[206]	Maximize system reliability; minimize the number of blocked hosts	NSGA	Resource allocation	Mobile ad hoc networks
[219]	Maximize energy savings; minimize the length of the schedule of task allocation	Energy-delay tradeoff algorithm	Task allocation	Network embedded systems
[224]	Maximize system reliability; Minimize resource consumption	Hybrid particle swarm optimization	Task allocation	Distributed systems
[231]	Maximize load balance among servers; minimize average response time for requests and delay	Service arrival based task allocation policy	Task allocation	Distributed systems
[233]	Minimize maximum latency in resource allocation (task-to-message; priority-to-task/message; signal-to-task)	Mixed integer linear programming; simulated annealing	Task allocation	Real-time distributed systems

specifically survey trust-based solutions which can be applied to all three classes. Further, we discuss the trends of hybrid approaches as MOO solution techniques in Section VI-E.

A. Class 1: Global Welfare Only

Class 1 covers MOO problems in which the system seeks to optimize multiple objectives. Table V summarizes the surveyed existing works belonging to Class 1. Below we first discuss key features in each work and then we summarize commonality and variability of formulation and solution techniques for Class 1 MOO problems.

Amin et al. [4] study the power loading problem for orthogonal frequency division multiplexing (OFDM) with imperfect channel estimation based on the tradeoff between energy efficiency and spectral efficiency. This work formulates the given problem as a multiobjective optimization problem to discuss the choices of the trade-off parameters under perturbations derived from system parameters.

Agarwal et al. [1] investigate a team formation optimization problem in an autonomous multi-robot system. This work aims to meet the dual conflicting objectives of maximizing both the task completion ratio and system efficiency in resource utilization using evolutionary genetic algorithms (i.e., NSGA, SPEA). Blake and Carter [22] propose a methodology to solve resource allocation of physicians to cases in hospitals using a goal programming method, where the two goals are to maintain physicians' income while minimizing disturbance to practice.

Balicki [11] studies a task assignment problem in a distributed environment based on a multi-objective quantum-based algorithm (MQEA). The objectives of this work are to maximize system reliability while minimizing workload and communication cost. Chen and Sun [38] study a task assignment problem in a multi-robot environment consisting of heterogeneous mobile robots, given resource constraints associated with tasks. They propose a coalition formation technique based on heuristic leader-follower structure that identifies optimal solutions of the task allocation problem based on the degree of energy required for task execution.

Bedoui et al. [19] solve resource allocation problem in broadcasting systems with the goals of minimizing hardware investment while maximizing QoS provided to users by proposing hybrid metaheuristics that combine NSGA-II with either SA or TS. Dib et al. [60] aim to identify a path minimizing time, cost, and distance in directed or undirected road networks. They use a hybrid metaheuristic based on genetic algorithm and variable neighborhood search, combing two metaheuristics. They conduct performance analysis by comparing the hybrid metaheuristics and exact algorithms such as Dijkstra's and integer programming.

Dieber et al. [61] solve the problem of identifying optimal configuration settings for a large-scale sensor network comprising camera nodes using an evolutionary algorithm. This work aimed to maximize quality-of-service (QoS) in the camera frame rate and resolution while using minimum energy and data volume. Dridi et al. [65] solve a resource allocation and scheduling problem for marine surveillance applications using a bi-colony ant-based approach. This work has two conflicting objectives: minimize job completion time while minimizing total cost. Guo et al. [88] examine a task assignment problem using a particle swarm optimization technique to meet multiple objectives that minimize task execution time and cost for data transfer between processors in cloud computing environments.

Jiang et al. [101] study the tradeoff between energy consumption and throughput as the conflicting system goals in a multihop wireless network. Jin et al. [104] use a genetic algorithm to obtain a task allocation that balances energy consumption by nodes for task execution and processing power provision in order to maximize network lifetime. Hajri-Gabouj [91] employs fuzzy logic and genetic algorithm to solve a task assignment problem, minimizing task execution time and maximizing load balance. Matsatsinis and Delias [134] consider a task to agent allocation problem, maximizing speed of task execution, minimizing resulting risk and achieving desired functionality. The authors used the ε-constraints solution technique for solving MOO.

Mezmaz et al. [137] present hybrid metaheuristics by combining GA and a memetic algorithm with a branch and bound algorithm to solve a flowshop scheduling problem for all jobs to be scheduled on multiple machines sequentially while minimizing the total completion time and the total tardiness. Milano et al. [142] propose a technique to represent system security in decentralized electricity markets in terms of voltage stability. The system has a multiobjective function that maximizes both social benefit and the distance to maximum loading conditions based on market-based pricing mechanisms.

Notario et al. [154] minimize consumption of bandwidth and energy and maximize quality of task execution based on a genetic algorithm. Shi and Bian [182] propose a solution for a business logistic alliance problem: maximizing profit margins and an importance degree of selected partners, using fuzzy logic and the concept of entropy. Shuai et al. [188] propose a generalized particle model in order to maximize the parallel optimization of resource allocation and task assignment/coordination in P2P and grid computing. Vidyarthi and Khanbary [206] study a channel allocation problem which aims to maximize reliability in data transmission and to minimize the number of blocked hosts using a NSGA.

Ouni et al. [158] propose a linear programming based optimization method to study the tradeoff between energy consumption and throughput in multi-hop wireless mesh networks using a MAC layer based on Spatial Time Division Multiple Access. Xie and Qin [219] propose an energy-efficient task assignment protocol based on the tradeoff between energy and delay to execute a task to minimize the length of schedules of task allocation and energy consumption. Yin et al. [224] study program modules to a processor allocation problem in a distributed environment which aims to maximize system reliability given resource constraints. This work uses hybrid particle swarm optimization to find close-to-optimal solution(s) but with a reduced complexity and solution search time.

Rachedi and Benslimane [165] study a multi-objective optimization problem with conflicting goals of security services (authentication, confidentiality, and integrity) and QoS (throughput, delay, and reliability) in wireless sensor networks. This work uses a multi-objective evolutionary algorithm, NSGA-II, and identifies different optimal security settings adapting to QoS and energy requirements.

Zhang et al. [230] take a Nash bargain-based game theoretic approach to identify solutions for the joint subchannel and power allocation for the uplink problem in cognitive small cell networks. They consider fairness among users and QoS (outage constraint, imperfect channel state information, and maximum power constraints) as system objectives. Zhang et al. [231] analyze the correlations between job arrivals in order to solve a task allocation problem for clustered systems. Their goals are to minimize the average response time (or delay) for received requests while maximizing load balance among servers. Zhu et al. [233] formulate a task allocation problem for distributed systems in order to minimize delay while meeting deadline constraints using *mixed integer linear programming* technique.

Commonality and Variability of Class 1 Works: For the works belonging to Class 1 in Table V, most of the works aim to solve MOO problems associated with resource or task allocation or coalition formation. The environments are mostly distributed systems including multi-robot systems, wireless sensor networks, P2P, or mobile ad hoc networks. Fig. 5 shows the summary of the MOO techniques used for 25 works listed in Table V where the works by [19, 60, 91, 137, 182, 233] use more than one techniques, respectively. In Fig. 5, a large volume of works use evolutionary algorithms (i.e., 34%) or bio-inspired algorithms (i.e., 12%) because most works aim to identify optimal solutions with a large size problem space. This requires searching for a large solution space, which is well considered in traditional metaheuristic optimization techniques.

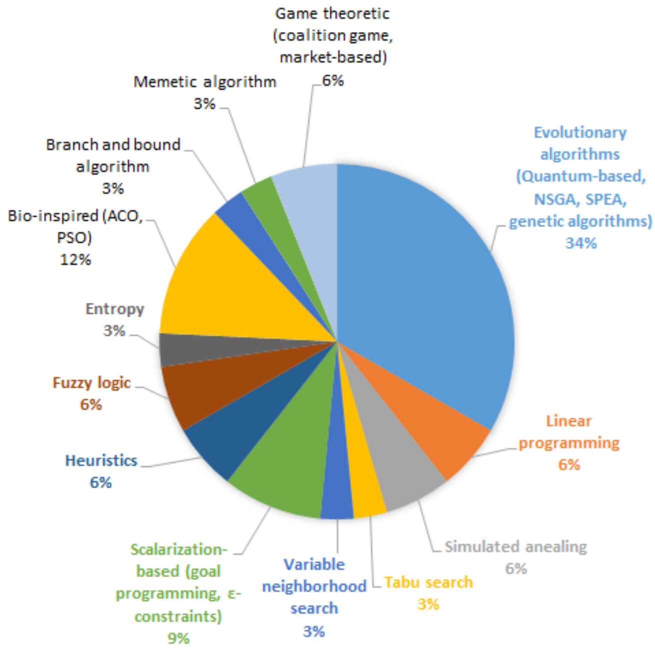


Fig. 5. MOO techniques used for Class 1

B. Class 2: Global Welfare and Individual Welfare with Identical Individual Payoff Functions

Class 2 covers MOO problems where all agents aim to achieve the same individual objective function. There is substantial literature on Class 2 MOO problems. Table VI summarizes the surveyed existing works belonging to Class 2. Below we first discuss key features in each work and then we summarize commonality and variability of formulation and solution techniques for Class 2 MOO problems.

Anagnostopoulos et al. [5] explore a solution for an online task assignment problem by matching a set of skilled people to each task. Their solution aims to minimize the communication overhead while balancing workloads. Analoui and Rezvani [6] use a microeconomic approach to solve a resource allocation problem in multicasting, treating a service as a good and an entity as a firm to provide goods. Their goal is to balance the service demand and resource supply and maximize the benefit of users.

Asl et al. [7] combine web services within communities based on a cooperative game to efficiently form coalitions of web services. Communities and web service providers aim to maximize their gains while the system aims to maximize fair-distribution of the gains among web service providers using the Shapley value. Brown et al. [31] solve a security problem using a game that deals with multiple security goals. They set a single goal based on the main objective while other objectives are treated as constraints. This work studies the impact of varying the constraints on the identified Pareto frontiers. Here the system objectives are based on the observed attacker type and each attacker, as a player, aims to maximize its payoff.

Cho et al. [42] solve a task assignment problem in tactical MANET environments that aims to meet multiple objectives such as minimizing communication cost incurred by task assignment while maximizing the ratio of task completion and nodes' utilization. Cai et al. [32] use a non-transferrable coalition formation game to solve a green uplink resource sharing problem between device-to-device (D2D) and cellular users. Their goal is to maximize resource utilization by optimizing mode selection and resource scheduling.

Das et al. [51] solve a service-based message sharing problem in vehicle ad hoc networks (VANETs) using a cooperative game with transferable utility and the concept of *core* for coalition formation. Each node aims to maximize the payoff by processing service-messages in a coalition that provides the highest incentives while each coalition maximizes its coalition payoff.

Dasgupta et al. [52] study a sailor-to-task assignment problem using NSGA in navy bases for minimizing the number of required sailors, while maximizing the total training and preference score with minimum cost. Deshpande et al. [59] use fuzzy set theory to solve a task assignment problem in multi-agent systems with two conflicting objectives: minimizing delay in task completion while maximizing the task completion ratio. Duan et al. [66] employ a cooperative game to design a rapidly converging sequential algorithm as a solution of large-scale application scheduling problem in hybrid cloud platforms. The system aims to maintain fairness of execution time to users while the users aim to minimize the execution time and cost by using ε-constraints.

Edalat et al. [68] propose an auction-based task assignment solution in wireless sensor networks with two global objectives: maximizing the overall network lifetime while satisfying application deadlines. An individual entity seeks to maximize its payoff by bidding on a task with low workload so as to consume less energy but have a high chance of being assigned to the task. Gao et al. [75] propose a cross-layer optimization method to solve a coalition formation problem in wireless sensor networks for minimizing energy consumption among sensors.

Garg et al. [77] use the Shapley value and coalition game theory to solve a clustering problem in data mining. This work has two goals: minimizing the mean distance from each point to the closest center and the mean distance of intra-cluster from point-to-point. Genin and Aknine [78] devise a coalition formation algorithm for multi-agent systems in which each agent selects a bundle of tasks (i.e., coalition) to join based on the similarity with previous bundles or occurrences of tasks in previous bundles. Each agent wants to maximize its utility by joining more bundles but with less resource consumption. A coalition, or a bundle, wants to maximize its payoff by completing all tasks in the bundle with current agents. Gharehshiran et al. [79] formulate a sub-channel allocation as a coalition formation problem in universal mobile telecommunication systems (UMTS) by considering it as a cooperative game in which an individual user wants to maximize its monetary payoff based on the network throughput while the system aims to maximize the overall network throughput. Guo et al. [89] use the VCG auction mechanism to propose a secure and efficient key management service for MANETs. They seek to maximize the ratio that the key management service is successfully provided and to

TABLE VI
EXISTING WORKS IN CLASS 2 WITHOUT USING TRUST

Ref.	Individual Objective	System Objective	Techniques	Problem	Domain
[5]	Users: Maximize utilization of skill sets; Minimizing workload and communication cost	Meet required skill sets; Minimize communication overhead; maximize load balance	Bi-criteria approximation	Team formation	Web-based social network
[6]	User: Maximize QoS utility	Network: Maximize QoS	Micro-economic theory	Resource allocation	Multi-service overlay networks
[31]	Attackers: maximize payoff	Defender: maximize resilience	Iterative ε-constraints; Integer linear program; Stackelberg game	Security strategy selection	Multi-agent systems
[42]	Minimize energy consumption; maximize task participation	Maximize mission completion ; Minimize communication overhead	Combinatorial auction theory	Task assignment	Mobile ad hoc networks
[52]	Cluster head: Meet three goals (energy consumption, load balance, and task deadline)	System: same goals as cluster heads	NSGA	Task assignment	Wireless sensor networks
[59]	User: Maximize QoS; minimize delay / communication cost	Maximize task completion; minimize delay	Fuzzy set approach; auction theory	Task assignment	Multi-agent systems
[68]	Minimize bidding delay	Minimize energy consumption / delay	Reverse auction	Task assignment	Wireless sensor networks
[75]	Minimize energy consumption	Maximize fair energy consumption	Cross-layer optimization	Coalition formation	Wireless sensor networks
[77]	Maximize payoff	Maximize coalition payoff	Cooperative game theory; Shapley value	Cluster formation	Clustering in data mining
[78]	Maximize node utilization	Maximize coalition payoff	Similarity and frequency based selection algorithms; Hedonic coalition game	Coalition formation	Multi-agent systems
[79]	User: Maximize network throughput	Maximize network throughput	Cooperative game theory	Coalition formation	Mobile telecommunication systems
[89]	Minimize cost in energy and security maintenance	Maximize successful key management service	VCG	Coalition formation for key management	Mobile ad hoc networks
[94]	Maximize the good with minimum resource consumption	Same objective as an individual agent but in a group level	Probabilistic model based on prior knowledge; dynamic programming	Resource allocation	Multi-agent system
[102]	Maximize task completion	Maximize task completion and load balance; minimizecommunication cost	Context-based heuristics	Task allocation	Complex software systems
[110]	Maximize payoff	Minimize false alarms; maximize throughput per cognitive radio	Coalition formation game; Markovian model	Coalition formation	Distributed cognitive radio networks
[116]	Minimize cost for queries recall and membership maintenance	Minimize convergence time to optimality, load balance, membership and communication cost	Cluster formation game	Formation of clustered overlays	Clustered overlay networks
[125]	Minimize resource consumption	Minimize resource consumption	Heuristic three phase algorithm	Task scheduling	Wireless sensor networks
[124]	Buyer: minimize price	Seller: maximize revenue	Combinatorial auction theory	Coalition formation	Online markets
[143]	Maximize payoff in file sharing	Maximize fairness and stability in cost sharing	Cooperative game theory; Shapley value	Coalition formation	P2P networks
[175]	Each division: maximize effective security resource usage; minimize security threats	Organization: same as division but in an organizational level	Coalition game theory	Resource allocation	IT resource management
[176]	Maximize payoff	Maximize throughput and valuation; minimize delay	Hedonic coalition game	Task allocation	Multi-agent wireless networks
[177]	Individual robot: maximize its task completion	Minimize maximum task completion cost; Minimize communication overhead	Heuristic multi-robot-cooperation framework	Task scheduling	Multi-robot systems
[184]	Robot: Minimize resource consumption	Maximize the total benefit of completed tasks; minimize delay	Genetic algorithm	Task assignment	Multi-robot systems
[183]	Maximize access time by secondary users	Minimize delay for coalition formation procedure	Vickery auction	Coalition formation	Cognitive ratio
[190]	Customer: maximize service satisfaction	Provider: maximize revenue	Nontransferable payoff	Resource allocation	Commercial wireless networks
[195]	User: Minimize cost and delay	Minimize workflows, task completion time, and communication overhead	Genetic algorithm	Task assignment	Cloud computing
[200]	Minimize energy consumption and task completion time; maximize task relevancy and priority	Maximize task completion; minimize delay	Auction theory	Multi-robots task allocation	Multi-robot systems
[215]	Minimize delay / communication cost	Maximize data transmission; minimize coalition cost	Tabu search	Coalition formation for routing	Wireless sensor networks
[217]	An agent: maximize utilization	Task planner: maximize task assignment	Consensus-based bundle algorithm	Task assignment	Multi-agent systems
[222]	Individual colony: minimize traversed distance	Minimize energy consumption and track errors	ACO	Task assignment	Multi-agent systems
[221]	User: Minimize cost	Minimize resource consumption; Maximize server utilization	NSGA	Resource allocation	Cloud computing
[225]	User: Meet deadline and budget requirements	Minimize execution time and cost	Genetic algorithms	Task assignment	Grid computing

minimize the nodes' energy consumption and cost to maintain security.

Guazzone et al. [86] take a hedonic game theoretic approach to form a cloud federation consisting of cloud providers (CPs) where each CP aims to reduce energy cost by joining a federation. The authors propose an algorithm to determine a federation set that maximizes the joint profit of the autonomous and selfish CPs while the federation maintains stability by meeting each CP's need to maximize its payoff. Hazon et al. [94] propose a resource coordination method based on a prior knowledge-based probability model for multi-agent systems executing exploration and patrol missions in order to maximize the average probability of acquiring the goods (e.g., battery charge) while minimizing the average resource consumption. Jiang and Jiang [102] consider a complex software system where agents are executing tasks with limited resources. They solve a task allocation problem using contextual information with the two goals of maximizing the task completion ratio and load balance with minimum communication overhead.

Khan et al. [110] model a cognitive radio network as a cooperative game and solve it as a coalition formation problem in order to pick cognitive radios that maximizes the overall throughput and minimizes the false alarm probability in detecting a primary user. Koloniari and Pitoura [116] use a game theory to solve a problem of cluster formation in clustered overlay network which is effectively used to exchange data appropriate received queries. A node decides to join a cluster based on its interest or the content of a cluster. In this problem, an individual node had two goals: minimizing the recall cost of the queries by having enough members in the same cluster (fetching data within a same cluster incurs less cost) and minimizing maintenance cost of having more memberships. The system goals were convergence speed to Pareto optimality, cost optimality in recall and membership maintenance, load balance, and cost minimization in dealing with nodes' movement and social cost.

Li et al. [125] propose a heuristic three-phase algorithm that minimizes energy consumption while maximizing task completion within a given deadline and energy consumption balance among individual sensor nodes. Li and Sycara [124] study coalition formation in an electronic marketplace where buyers bid on multiple items based on combinatorial auction offering reservation costs while sellers can offer discount prices depending on the volume ordered by the buyers. They developed a polynomial time algorithm to identify an optimal coalition of buyers per item that maximizes the revenue on the item. Militano et al. [143] deal with a file sharing problem aiming to maximize fairness and stability among users in P2P networks. The problem was modeled as a "cooperative game" where an individual user wants to minimize expense but maximize the file sharing time based on the Shapley value.

Mashayekhy and Grosu [133] propose a solution to form a virtual organization (VO) in grid computing platforms using a coalitional game. The proposed mechanism allows the grid service providers (SPs) to maximize their payoff by joining an organization with the highest benefit (i.e., efficient resource utilization and minimum delay) while VO also maximizes its payoff by maintaining group stability which prevents the SPs from leaving VO. Pillai and Rao [163] take a game theoretic approach to solve a resource allocation problem among virtual machines on a cloud where forming a coalition of virtual machines is needed for efficient resource utilization. The system aims to minimize job completion time and wastage of resource while each coalition maximizes request satisfaction.

Sariel-Talay et al. [177] propose a distributed, energy-efficient heuristic cooperative task scheduling framework for multi-robot systems. Each robot seeks to maximize its payoff as a share of the coalition payoff while the system seeks to minimize maximum task completion time and communication overhead. Saad et al. [175] introduce coalition game for "security risk management." They investigated how autonomous divisions can cooperate with interdependent security vulnerabilities and assets. An organization consisting of multiple divisions wants to maximize effective security resource while minimizing security threat. Saad et al. [176] study task allocation problem in wireless networks using hedonic game theory with the system objectives: maximizing throughput and minimizing the average delay in task execution. Each agent wants to join a coalition to maximize its payoff as a share of the coalition payoff. An agent can freely select any task whenever tasks are available. However, communication cost was not considered when an agent switches from one task to the other.

Shi et al. [184] solve a task allocation problem for multi-robot systems using decision preference information as a factor of the fitness function in a genetic algorithm. This deals with multiple objectives where a robot aims to save energy while the system seeks to maximize task completion with minimum delay. Shi et al. [183] propose a cooperative sensing protocol for secondary users using Vickery auction in cognitive networks. The cooperative auction is advanced by prioritizing access rights while decreasing response time for coalition formation process.

Singh et al. [190] investigate a resource allocation problem in a commercial wireless network where the providers' objective is to maximize their revenue by providing wireless network services while the customers' objective is to maximize their satisfaction from the service received. This work uses a nontransferable payoff, showing that cooperation among providers is the best strategy to maximize payoffs of both providers and customers. Song et al. [191] solve a resource allocation problem in cellular networks where device-to-device (D2D) communication must meet multiple objectives in radio resource allocation. In the proposed scheme, the system aims to minimize transmission power of D2D users while each user has the goal of maximizing the utility of the radio resource.

Szabo et al. [195] examine a task allocation problem in cloud computing using evolutionary genetic algorithms. Here the system goals are to minimize workflows, delay introduced by the task completion, and communication cost while each individual user wants to minimize cost and service delay. Tolmidis and Petrou [200] propose an auction theoretic approach to solve a task allocation problem in multi-robot systems where each robot is able to perform several functions. An individual robot has the goals to minimize energy consumption and delay in task completion while maximizing the degree of relevancy and priority level to an assigned task. Similarly, the system aims to maximize completed tasks and minimize delay introduced due to task assignment and completion.

Wang et al. [210] consider overlapping coalition formation for resource allocation in a distributed cooperative sensing system. The proposed scheme allows secondary users to join multiple overlapping coalitions to maximize the use of radio resource while each coalition maximizes its payoff representing the maximization of its member sensing performance. Weerakoon and Allan [215] propose an energy-efficient autonomous data routing scheme for wireless sensor networks, treating selection of transmitters as a coalition formation problem. They use a heuristic decentralized mechanism based on Tabu search in order to identify a coalition maximizing data transmission with minimum cost for coalition formation. Whitten et al. [217] propose a decentralized algorithm to solve task assignment problem in that each entity can autonomously make decisions, given constraints. They used a bundle algorithm based on consensus in order to model different behavioral strategies. This work considers the same goals to maximize task assignment performance in formulating the objectives of a task planner and each agent.

Xu et al. [222] take a bio-inspired multiple ant colony approach to solve a task assignment problem in multiple unmanned underwater vehicles in order to minimize both the total distance multiple vehicles traveled and the total turning angles to minimize energy consumption and track errors. Xu et al. [221] solve a multi-objective task assignment problem using NSGA in cloud computing environments. They seek to minimize the total power consumption in virtual machines and maximize useful utilization of each server. Yu et al. [225] solve a multi-objective workflow execution problem in utility Grids modeled as a Directed Acyclic Graph. This work considered the two objectives of minimizing execution time and cost (i.e., price), using three genetic algorithms including NSGA, SPEA, and PAES. Xu et al. [220] propose a network selection algorithm for vehicular networks by using a cloud's rich computing and data storage resources. The authors model a network selection process as a coalition formation problem based on hedonic game theory so that a vehicle can choose a network that best provides high quality, uninterrupted service while the system aims to maximize throughput and fairness in vehicles' utilities.

Commonality and Variability of Class 2 Works: The commonality of MOO research in Class 2 is that payoffs earned by individual entities often directly or indirectly contribute to global welfare. Therefore, in Class 2 MOO research, an individual entity's goal is aligned with that of a coalition in that they aim to be mutually beneficial to maximize their payoffs. Based on the summary shown in Table VI, the main application domains for Class 2 are distributed environments including multi-agent systems, mobile ad hoc or sensor networks, or cloud computing which are not much different from those for Class 1. In addition, the main MOO problems are task allocation or scheduling, coalition / cluster formation, or resource allocation, which are not much different from those observed in Class 1 as well. Comparing Table VI with Table V, we observe that a large amount of works use game theoretic approaches. Fig. 6 gives the overall key trends of the main techniques used in Class 2 based on 32

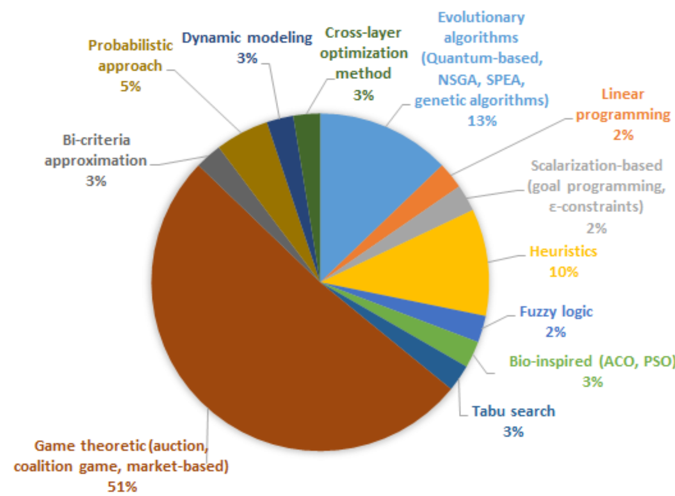


Fig. 6. MOO techniques used for Class 2

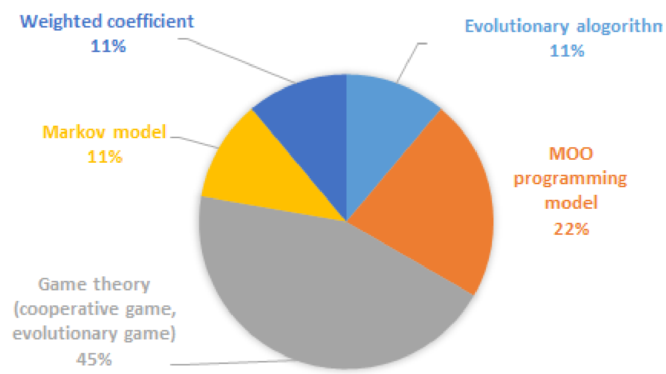


Fig. 7. MOO techniques used for Class 3

existing works in Table VI. Some works use more than one techniques (i.e., three techniques in [31] and two techniques in [61, 79, 95, 111, 144]). The reason for the popularity of game theoretic approaches derives from the problem nature in that multiple parties aim to optimize their objectives where game theoretic approaches can provide useful modeling tools.

C. Class 3: Global Welfare and Individual Welfare with Different Individual Payoff Functions

Class 3 covers MOO problems that deal with different individual payoff functions for situations in which individuals may want to maximize different payoff functions because different customers may have different user satisfaction criteria towards a service provided. This scenario is rare because most existing studies dealt with objectives from an individual entity vs. a system (or coalition). Nevertheless it is a new trend worthy of attention. Table VII summarizes the surveyed existing works belonging to Class 3. Below we first discuss key features in each work and then we summarize commonality and variability of formulation and solution techniques for Class 3 MOO problems.

TABLE VII
EXISTING WORKS IN CLASS 3 WITHOUT USING TRUST

Ref.	Individual Objective	System Objective	Techniques	Problem	Domain
[15, 16]	Users: maximize respective utility	Maximize the overall throughput and transmit power	MOO programming based on pre-defined interference thresholds of primary users	Resource allocation	Cognitive radio systems
[17]	Users: maximize respective utility	Maximize the overall throughput and transmit power	Linear weighted coefficient	Resource allocation	Cognitive radio systems
[18]	Users: maximize respective utility	Maximize the overall throughput and transmit power	Evolutionary algorithms	Resource allocation	Cognitive radio systems
[96]	Maximize the objective function of an individual firm	Maximize the overall objective function	MOO programming model	Optimal selection of alliance partners	Business supply chain alliance
[130]	Maximize an individual payoff	Maximize the overall group payoff	Coalitional game theory; Markov process	Clustering	Clustering in data mining
[135]	Maximize an individual player's objective where each individual has a different objective	Minimize computational cost; Maximize optimality accuracy	Cooperative game theory; evolutionary game	Generic MOO solution	Computational modeling technique
[153]	Maximize an individual user's QoS satisfaction level	Maximize the satisfaction of QoS constraints while minimizing the impact of global QoS caused by new arrivals of services	Heuristic cooperative coalition formation model	Coalition formation	Distributed systems
[194]	Users: maximize the satisfaction of QoS	Maximize QoS based on load balance among service providers	Cooperative game theory; Nash bargaining solution	Task allocation	Grid computing

Bedeer et al. [15] solve the problem of both maximizing the throughput and minimizing its transmit power based on the constraints of both secondary and primary users in the optimal link adaptation problem of orthogonal frequency division multiplexing-based cognitive radio systems. In their MOO problem, two users, primary and secondary users, aim to maximize their respective objectives for optimal power and bit allocation in cognitive radio systems [16, 17]. The same authors [18] extend their work by using evolutionary algorithms to deal with non-convex constraints.

Huang et al. [96] provide a strategic coalition formation solution in supply chain alliance where independent firms choose their partners to exchange or share resources. An individual firm aims to maximize the sum of its own satisfaction in profit, service quality and customer satisfaction by using an interdependent multi-objective programming technique. Liu et al. [130] take a game theoretic approach using the Shapley value to solve a clustering problem in data mining. The authors consider a multi-objective categorization problem modeled as a coalition game and analyze it based on a Markov process model for maximizing the payoffs of each coalition and individual family. Meng et al. [135] transform a multi-objective problem into a game with multiple players based on the proposed generic mathematical method.

Nogueira and Pinho [153] study a team coordination problem in resource constrained distributed systems based on coalition game theory. The selected team provides services based on collaboration with qualified neighbors to maximize the satisfaction of QoS and to minimize the impact of new service request arrivals on the global QoS. Subrata et al. [194] consider a service allocation problem in grid computing. When a user sends a task to a server, a set of service providers cooperate to service the received request with an acceptable QoS. The authors model this as a cooperative game using the concept of Nash Bargaining in order to best balance the profits of all users while meeting multiple QoS objectives.

Commonality and Variability of Class 3 Works: Compared to the works in Class 1 and Class 2, there are fewer works dealing with the characteristics of objectives in Class 3; so we summarize only 8 works in Table VII. For the main MOO techniques in Class 3, as shown in Fig. 7, we see the dominant usage of game theoretic approaches as Class 3 has multiple objectives associated with both system and individual user perspectives. Note that [130] uses two techniques while all other works use a single technique.

D. Trust-Based Approach Across Three Classes

As discussed in Section IV-D, trust-based approaches have been used to find close-to-optimal solutions as the approximation of optimal solutions with low complexity. In this section, we discuss trust-based heuristic MOO approximation solutions based on the publications since 2000 across the three classes.

a) Trust-based MOO approach in Class 1: Das and Islam [50] propose a computation model dealing with dynamic trust that detects malicious nodes but maintains high load balance among nodes. Trust is measured based on multiple dimensions such as feedback credibility, service satisfaction, and preference similarity. Trust is used as a proxy to estimate the degree of QoS that can be provided by an agent to other agents in a multi-agent system. Dorn et al. [64] propose a team formation algorithm using genetic algorithm and simulated annealing to compose a team with qualified members who have a required skill level and maintain trust relationships with others. Shen et al. [181] develop a trust-based solution for job scheduling in grid management with multiple system objectives including security, reliability, load balance and throughput. Ye et al. [223] solve a trustworthy worker selection problem in a crowdsourcing environment as a multi-objective combinatorial problem. This work enhances NSGA-II by using trust estimates based on context-based trust, considering task type and reward amount.

b) Trust-based MOO approach in Class 2: Chang et al. [36] propose a task assignment algorithm based on composite trust in order to choose members that maximize their utilization and mission completion ratio but maintain an acceptable risk level based on the auction theory. Cho et al. [40] extend [36] by considering additional system goals such as minimizing task delay and resource consumption. They validate the optimality of trust-based solutions via comparisons with the solution provided by integer linear programming. Goel and Stander [84] present a clan formation algorithm based

TABLE VIII
EXISTING WORKS USING TRUST ACROSS CLASSES

Ref.	Individual Objective	System Objective	Techniques	Problem	Domain
Class 1					
[50]	N/A	Maximize accuracy of trust assessment to provide high QoS and load balance	Trust computation model called SecureTrust	Resource allocation	Multi-agent systems
[64]	N/A	Maximize skill coverage and team connectivity	Genetic algorithms; simulated annealing	Team formulation	Web-based social networks
[181]	N/A	Maximize security, reliability, load balance and throughput	Trust-driven grid job scheduling	Resource allocation	Grid computing
[223]	N/A	Maximize accuracy of selecting trustworthy workers	NSGA-II; context-based trust based worker selection	Combinatorial worker selection	Crowdsourcing
Class 2					
[36]	Maximize its own payoff by minimizing energy consumption but maximizing task participation	Maximize task completion ratio	Auction theory	Task assignment	Mobile ad hoc networks
[40]	Maximize trust-based payoff	Maximize mission completion; minimize resource consumption and task delay	Integer linear programming; trust-based auction	Task assignment	Coalition networks
[84]	Maximize payoff as a share of the coalition payoff	Maximize coalition payoff by minimizing missed cooperation opportunities; Minimize communication overhead for clan formation	Cooperation-based clan formation	Clan formation	Multi-agent systems
[87]	Migrating worker: Maximize membership period by selecting the best workplace	Workplace: Maximize efficiency and security in business process in terms of time and resource	Trust and self-confidence based coalition formation	Coalition formation	Business workflow management
[97]	Partner: Maximize payoff as a share of the alliance payoff	Alliance: Maximize the profit of the supply chain alliance	Cooperative game	Alliance formation	Business supply chain alliance
[129]	Customers: Minimize prices for resources under constraints	Providers: Maximize resource sharing and security	Economic-driven allocation; Q-learning	Resource management	Grid computing
[141]	Maximize a payoff as a share of the coalition payoff	Maximize trust synergy; minimize trust liability	Cooperative game	Coalition formation	Multi-agent systems
[150]	Maximize a payoff as a share of the coalition payoff	Maximize coalition payoff	Spatial prisoner's dilemma game	Coalition formation	Multi-agent systems
[208]	Minimize energy consumption; maximize load balance	Maximize mission completion based on security and reliability criteria; minimize mission completion time	Heuristic trust-based min-min algorithm	Task scheduling	Mobile ad-hoc grid computing
[229]	Join a group with high trust	Maximize trust in a task-oriented social network while minimizing communication cost	Heuristic trust-based algorithm	Team formation	Web-based social network
[213]	Maximize payoff	Maximize mission reliability; minimize task delay and utilization variance	Weighted sum; auction theory; trust-based member selection	Task assignment	Service-oriented mobile ad-hoc networks
Class 3					
[30]	Vendor: Maximize sales; Customer: Minimize prices	System: Maximize stability of an optimal formation of coalitions	Trust-based coalition formation	Coalition formation	Multi-agent systems
[209]	Maximize preference-based QoS	Maximize QoS based on user preference and trust	Weighted sum based on preference and trust	Web service selection	Web services

on trust and motivation in which an agent is self-interested to maximize its payoff in a clan. The goal of a coalition is to maximize payoff by losing cooperative opportunities with minimum communication cost in forming a clan.

Guo et al. [87] propose a trust and self-confidence based coalition formation scheme to migrate workflow to ensure security (e.g., a worker's integrity) and efficiency (e.g., a worker's experience) in executing business processes where the degree of cooperation, as a dimension of trust, is estimated based on an entity's capability and security. Huo et al. [97] study how to select alliance partners to maximize the profit of the supply chain alliance. This work models the alliance partner selection process based on game theory where an agent and an alliance party use trust to form a supply chain alliance with the goal of maximizing the profit of the alliance. Each agent aims to maximize its payoff as a share of the alliance's profit.

Lin and Huai [129] incorporate trust into resource management decision in grid environments to meet efficiency of resource sharing and to maintain security by correctly detecting malicious entities. In this market-driven resource sharing environment, consumers aim to minimize the price for using resource in completing tasks under budget and deadline restriction. A provider should form a club for customers that maximizes the total revenue and resource utilization by allowing only trustworthy entities to maximize effectiveness of resource usage.

Mikulski et al. [141] propose a theoretical coalition formation framework that uses trust relationships in multi-robot systems. This work uses cooperative game theory with a coalition aiming to maximize its payoff which can be estimated by the trust levels of members. Each agent also wants to maximize its payoff by joining a coalition with highly trustworthy members. As system objectives, this work aims to maximize trust synergy while minimizing trust liability as a proxy to measure the capability of a mission team, in which each member has an objective to maximize its payoff as a share of the coalition. However, no specific trust metric is discussed.

Nardin and Sichman [150] present a coalition formation solution using the spatial prisoner's dilemma game and analyzed the effect of trust on coalition dynamics. Like other existing work on coalition formation, this work models both a coalition's goal and an agent's goal as maximizing its payoff

TABLE IX
HYBRID MOO APPROACHES ACROSS CLASSES

Ref.	Individual Objective	System Objective	Techniques	Problem	Domain
Class 1					
[1]	N/A	Maximize both the number of completed tasks and task efficiency	NSGA; SPEA	Coalition formation under a single task	Multi-robot systems
[91]	N/A	Minimize task execution time and resource consumption	Fuzzy logic, genetic algorithm	Task assignment	Manufacturing systems
[224]	N/A	Maximize system reliability; Minimize resource consumption	Hybrid particle swarm optimization	Task allocation	Distributed systems
[233]	N/A	Minimize maximum latency in resource allocation (task-to-message; priority-to-task/message; signal-to-task)	Mixed integer linear programming; simulated annealing	Resource Allocation	Real-time distributed systems
[64]	N/A	Maximize skill coverage and team connectivity	Genetic algorithms; simulated annealing	Team formulation	Web-based social networks
[223]	N/A	Maximize accuracy of selecting trustworthy workers	NSGA-II; context-based trust based worker selection	Combinatorial worker selection	Crowdsourcing
[137]	N/A	Find a schedule of a set of jobs on a set of machines that minimizes the total completion time and the total tardiness	Hybrid metaheuristics combining GA and memetic algorithm with a branch and bound algorithm	Flowshop scheduling problem	Manufacturing production systems
[60]	N/A	Identify a path minimizing cost, time, and distance in directed or undirected networks	Hybrid metaheuristics with GA and VNS	Optimal path identification	Road networks
[19]	N/A	Maximize QoS to users; minimize hardware investment	Hybrid metheuristics combining NSGA-II with either TS or SA	Resource allocation	Broadcasting system
Class 2					
[59]	User: Maximize QoS; minimize delay and migration overhead	Maximize task completion with minimum delay	Fuzzy set approach; auction theory	Task assignment	Multi-agent systems
[77]	Maximize the share of coalition payoff	Maximize the coalition payoff	Cooperative game theory; Shapley value	Cluster formation	Clustering in data mining
[94]	Agent: Maximize the good with minimum resource consumption	Group: same objective as an agent but in a group level	Probabilistic model based on prior knowledge; dynamic programming	Resource allocation	Multi-agent systesm
[110]	Maximize payoff	Minimize false alarms; maximize throughput per cognitive radio	Coalition formation game; Markovian model	Coalition formation for cognitive radio	Distributed cognitive radio networks
[124]	Buyer: minimize price	Seller: maximize revenue	Combinatorial auction theory	Coalition formation	Online markets
[143]	Maximize payoff in file sharing	Maximize fairness and stability in cost sharing	Cooperative game theory; Shapley value	Coalition formation	P2P networks
[40]	Maximize trust-based payoff	Maximize mission completion; minimize resource consumption and task delay	Integer linear programming; trust-based auction	Task assignment	Coalition networks
[87]	Migrating worker: Maximize membership period by selecting the best workplace	Workplace: Maximize efficiency and security in business process in terms of time and resource	Trust and self-confidence based coalition formation	Coalition formation	Business workflow management
[129]	Customers: Minimize prices for resources under constraints	Providers: Maximize resource sharing and security	Economic-driven allocation; Q-learning	Resource management	Grid computing
[213]	Maximize payoff	Maximize mission reliability; minimize task delay and utilization variance	Weighted sum; auction theory; trust-based member selection	Task assignment	Service-oriented mobile ad-hoc networks
Class 3					
[130]	Maximize an individual payoff	Maximize the overall group payoff	Coalitional game theory; Markov process	Clustering	Clustering in data mining
[135]	Maximize an individual player's objective where each individual has a different objective	Minimize computational cost; Maximize optimality accuracy	NE, cooperative, and evolutionary game	Generic MOO solution	Computational modeling technique
[194]	Users: maximize the satisfaction of QoS	Maximize QoS based on load balance among service providers	Cooperative game theory; Nash bargaining solution	Task allocation	Grid computing
[209]	Maximize preference-based QoS	Maximize QoS based on user preference and trust	Weighted sum based on preference and trust	Web service selection	Web services

where an individual agent's payoff is a share of the coalition payoff.

Wang et al. [208] propose a trust-based heuristic task scheduling technique for mobile ad hoc grid computing environments for maximizing mission completion ratio based on required levels of security and reliability in task assignment and minimizing delay to mission completion. Zhan et al. [229] study team formation in online task-oriented social network applications that maximize trust values of entities with minimum communication cost where an individual member wants to join a group with maximum trust while the team aims to maximize the overall trust of all members.

Recently Wang et al. [213] propose a trust-based task assignment protocol to maximize reliability while minimizing task delay and utilization variance in the member selection process for task assignment in a service-oriented ad hoc networks. This work uses a weighted sum to consider a system's multiple objectives where each node aims to maximize its utility to task completion based on auction process.

c) Trust-based MOO approach in Class 3: Breban and Vassileva [30] investigate a coalition formation problem where vendors and customers are in long-term relationships where a vendor or a customer can join a coalition to maximize respective payoff based on assessed trust and the system wants

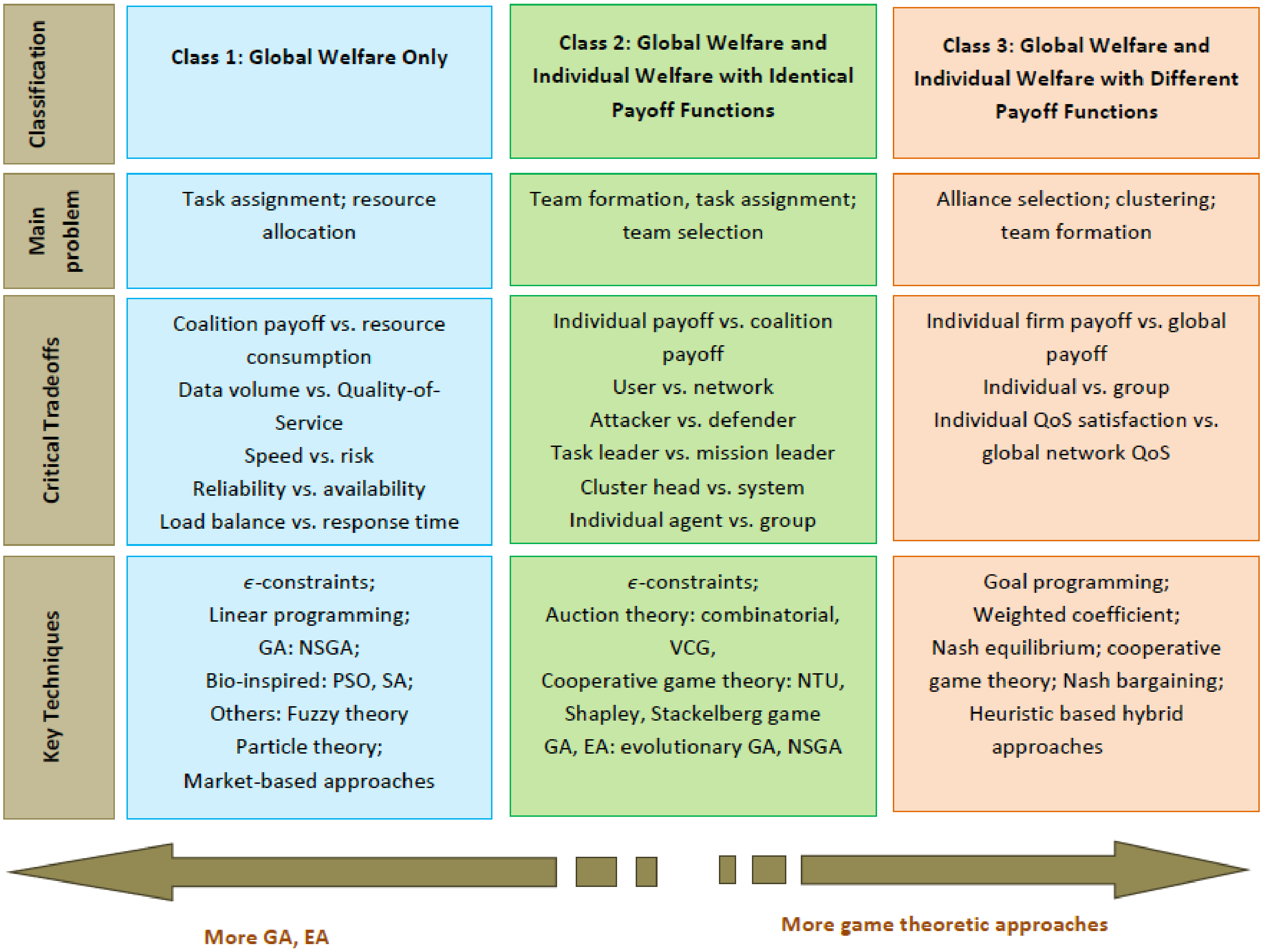


Fig. 8. The trends of the three classes of MOO Problems

to quickly reach an equilibrium state in coalition formation. In this work, trust is evaluated based on positive transaction experience and preference similarity. Wang et al. [209] formulate a service selection problem as a MOO problem to maximize QoS of users based on their preferences while multiple service providers aim to maximize QoS provided to users by selecting relevant services based on a user's preferences and trust.

E. Hybrid Approach Across Three Classes

Figure 8 summarizes the major trends observed in the three classes of MOO problems and solution techniques. The main problems are common among the three classes. The critical tradeoffs show the difference among the three classes of how the parties want to maximize their utilities. Among key solution techniques, the major trend is that Class 1 uses more traditional approaches including EA, bio-inspired algorithms (i.e., ACO, PSO) or scalarization-based techniques while Classes 2 and 3 use more game theoretic approaches. The reason is that MOO problems in Classes 2 and 3 aim to meet objectives of multiple parties while Class 1 only aims to maximize system objectives.

Based on our survey, we found many works use various types of hybrid approaches to solve MOO problems in diverse domain contexts. As shown in Table IX, heuristic-based approaches (e.g., trust-based, preference-based, confidence-based) are used in developing fitness functions in evolutionary algorithms [223] or simply combined with scalarization-based MOO techniques (e.g., weighted sum, linear programming) [40, 209, 213]. Further, cooperative game theory is commonly used with other game theoretic algorithms (e.g., Shapley value or Nash bargaining) [77, 143, 194]. Other works take hybrid approaches by combining game theory and evolutionary algorithms [135] or probabilitic theory and game theory [110, 130]. Note that hybrid metaheuristics include hybrid approaches combining metaheuristics with another method, either metaheuristics or other non-metaheuristics algorithms. On the other hand, the "hybrid approach" discussed here refers to any combination of more than one algorithms to solve MOO problems regardless of whether it uses hybrid metaheuristics or not.

VII. Discussions of Issues and Challenges

In this section, we discuss the key issues and challenges when solving MOO problems. In particular, we discuss the issues and challenges associated with dealing with MOO under uncertainty, the natures and properties in solving MOO related to Pareto optimality, duality, and solvability. In addition, we discuss the stability of optimal solutions and tradeoffs between optimality and complexity (i.e., scalability) in solving MOO problems.

A. MOO under Uncertainty

MOO problems have been solved by various types of multi-objective evolutionary algorithms (MOEAs) which are known to be powerful tools to solve global optimization problems with deterministic problem functions. However, in reality, uncertainty derived from incorrect system models and/or dynamic environmental factors hurts the accuracy of identified MOO solutions [128]. In solving most engineering problems under uncertainty, two types of uncertainty are often discussed: *aleatory uncertainty* and *epistemic uncertainty*. Aleatory uncertainty is derived from the randomness of underlying phenomenon representing variability of observations. On the other hand, epistemic uncertainty refers to the uncertainty associated with incorrect system models caused by insufficient or incomplete knowledge towards a real-world [128]. Since aleatory uncertainty is data-based and may not be reduced or modified, there has been some effort to reduce epistemic uncertainty [128].

Jin and Branke [105] categorize four causes of uncertainties in evolutionary computation as follows:

1) *Noise in fitness functions*. The noise may come from measurement errors or randomness. A typical noise model is given by:

$$\begin{aligned} F(X) &= \int_{\infty}^{-\infty} [f(X)+z]p(z)dz \qquad (17) \\ &= f(X), z \sim N\Big(0, \sigma^2\Big) \end{aligned}$$

where X is a vector of design variables, $f(X)$ is a time-invariant fitness function, $p(z)$ is a probability distribution of disturbance z, and z is additive noise. z is often assumed to follow a Gaussian distribution with zero mean and variance σ^2. Uncertainty is often introduced by the following two aspects: (1) in reality, the dynamics of environmental factors may affect the fitness function which should vary over time and noise z may not follow a Gaussian distribution; and (2) for efficiency, expected fitness functions often are used to approximate random samples.

2) *Robustness*. After optimal solutions are identified, changes in design variables or environmental factors may impact the accuracy of the identified optimal solutions. If the found solutions do not work under the changed environments, then the solutions are not robust to system and environmental dynamics. A robust solution implies robustness under high system and environmental dynamics by generating high performance based on the solutions from the robust fitness function. However, a robust solution may not be optimal. When noise is not avoidable, an individual solution cannot be accurately evaluated. Therefore, there often exists a tradeoff between accuracy and robustness of solutions.

3) *Fitness approximation*. If the fitness function cannot be expressed analytically, or if it is too expensive to compute, then approximated fitness functions often are used, accordingly leading to inaccuracy. Therefore, there exists a tradeoff between cheap but inaccurate solutions and expensive but accurate solutions when using an original fitness function or an approximated fitness function.

4) *Time-varying fitness functions*. Uncertainties may be introduced while time-varying fitness functions need to keep track of previous optimal solutions for computing a current optimal solution under changed environments over time.

In solving MOO problems with multiple and expensive fitness functions, a solution with high fitness variance may be treated as robust. To reduce uncertainties from this high variance, it is a better choice to consider a fitness function and a variance for each objective which allows searching for improved solutions with different tradeoff points between accuracy and robustness [105]. Recently Iancu and Trichakis [100] propose a methodology to maintain both robustness and Pareto optimality generating the so called *Pareto robustly optimal solutions*.

Some efforts at reducing uncertainties have been made in terms of theoretical analysis and network applications. Fieldsend and Everson [72] propose a method called *probabilistic domination contours* (PDC) to consider the confidence placed on objectives to be optimized under uncertainty. The authors conduct experiments to investigate the effect of the PDC; but the assumption of Gaussianity and independent noise may be too restrictive for many problems.

To solve MOO problems under dynamic system and environmental conditions, robust optimization functions are proposed in the literature [106, 127]. Johnston [106] proposes an evolutionary algorithm to solve a multi-objective scheduling in a deep space network, NASA's collection of assets communicating with spacecraft beyond near-earth orbit. Liao et al. [127] propose a method to solve a mobility robustness optimization problem in LTE self-organizaing networks in which a set of non-convex functions have conflicting goals. However, as discussed earlier, there exists an inherent tradeoff between optimality and robustness of fitness functions which are not addressed in these works.

B. Nature and Properties of Objectives and Solutions

In this section, we discuss the key factors in MOO and discuss the interplay between them in terms of Pareto optimality and solution efficiency in the following aspects: (1) Pareto optimality conditions based on Karush-Kuhn-Tucker theorem; (2) duality; (3) solvability; and (4) stability. There is a rich volume of literature [20, 56, 74, 80, 85, 139, 149, 178, 216] that discusses the relationships between the properties. Focusing on the scope of this work, we briefly discuss some key points.

1) Pareto Optimality Conditions: Karush-Kuhn-Tucker (KKT) theorem provides conditions to find Pareto optimal solutions in terms of necessary conditions and sufficient conditions [56]. Since the in-depth discussion of this theorem is very broad and has been published in a large volume of literature, we capture only the core aspects of this theorem. The KKT conditions are

$$\sum_{m=1}^{M} \lambda_m \nabla f_m(\mathbf{x}^*) - \sum_{j=1}^{J} u_j \nabla g_j(\mathbf{x}^*) = 0, \text{and} \quad (18)$$
$$u_j g_j(\mathbf{x}^*) = 0 \text{ for any } j$$

where M is the set of objectives, J is the set of constraints, and $\mathbf{x}^*$ is the set of solutions. λ and $\mathbf{u}$ are the vectors of constants. $f_i(\cdot)$ is the i-th objective function and $g_j(\cdot)$ is the j-th constraint function. Based on the conditions in Eq. (18), we discuss necessary and sufficient conditions for a solution to be *weak Pareto optimal* as below:

- *Necessity*: A necessary condition for a solution $\mathbf{x}^*$ to be weak Pareto optimal is that vectors $\lambda > 0$ and $\mathbf{u} \geq 0$ exist in which $\lambda \in \mathbb{R}^M$, $\mathbf{u} \in \mathbb{R}^J$ and λ, $\mathbf{u} \neq 0$ such that the conditions in Eq. 18 are true. Meeting these conditions does not necessarily guarantee the existence of weak Pareto optimal solutions [56].
- *Sufficiency*: Given that objective functions are convex and constraint functions are non-convex where both functions are differentiable at solution $\mathbf{x}^*$, a sufficient condition for $\mathbf{x}^*$ to be weak Pareto optimal is that $\lambda > 0$ and $\mathbf{u} \geq 0$ exist in which $\lambda \in \mathbb{R}^M$, $\mathbf{u} \in \mathbb{R}^J$ such that the conditions in Eq. (18) are true.

Pareto optimality is not always found when objective functions are not convex or objective / constraint functions are not differentiable at $\mathbf{x}^*$. Naturally, more and/or diverse types of objective / constraint functions may lead to no feasible solution region. Many approximation methods relax conditions for objective or constraint functions aiming to identify feasible solutions with the sacrifice of optimality. Due to the space constraint, we refer interested readers to [56, 178] for the proofs.

2) Duality: Duality provides two perspectives to solve a given optimization problem. An original problem, called a *primal problem*, often is converted to another problem, called a *dual problem*, to find lower bounds to the optimal solutions of the primal problem [149]. The reason for solving the dual problem is because of the significant computational advantages [149]. However, the dual problem does not necessarily generate the same optimal solution as the primal problem in which the difference is called *duality gap*. If the duality gap is zero, it is called *strong duality*. If the duality gap is nonnegative, it is called *weak duality* [149]. Two well-known dualities are *Lagrange duality* and *conjugate duality* [178]. As the broad discussion of these two dualities is out of scope of this work, interested readers are referred to [178].

For multiobjective linear problems, the relationship between duality and the existence of Pareto optimality has been studied [20, 139, 216]. Their studies prove that if a duality gap exists, the optimal solutions $\mathbf{x}^*$ from the auxiliary problem in a given linear problem are Pareto optimal [20, 216]. However, this effort has been made only in solving linear problems where objective functions are convex. Many other studies also examine the existence of weak Pareto optimality in convex, differentiable MOO problems [57, 58, 95].

3) Solvability: Although the assumption of the convexity of objective functions in MOO can significantly help complexity of optimal solution search, it is not a realistic assumption in practice [74]. The gap between theoretical MOO analysis assuming convex objective functions and real-world problems showing non-convex objective functions has been realized by many researchers [74]. Many MOO problems are known to be NP-Hard [74]. Recently the optimization research community discussed that the complexity of a MOO version of a combinatorial single objetive problems is NP-hard [85]. To relax NP-Hardness of MOO problems, many approximation techniques have been proposed and analyzed in terms of optimality accuracy (e.g., distance from optimal solutions to approximated solutions) and complexity (e.g., efficiency introduced by approximation techniques) [80]. As discussed in Section IV, many heuristic approaches have been proposed to mitigate the computational complexity of optimal solution search. However, due to the challenges in finding the ground truth optimal solutions, even validating the effectiveness of heuristic-based approximation mechanisms remains limited to some extent in practice.

C. Stability of Optimal Solutions

Stability analysis examines if identified optimal solutions are stable within a small change under perturbation [227]. Stability analysis is a useful tool for the post-analysis of identified optimal solutions. More specifically, it finds the lower and upper semicontinuity functions of optimal values and optimal solution sets [178]. For the stability analysis of MOO problems, Sawaragi et al. [178] define the MOO problems based on the following two parameters: (1) *parameter* u varies over a set of U specifying the set of feasible solutions; and (2) *parameter* v varies over a set of V describing the domination structure of the decision maker in the objective space. Let $X(u)$ be the set of all feasible solutions and $f(x, u)$ be the vector-valued function. The set of feasible solutions in the objective space, $Y(u)$, is

$$Y(u) = \{y \in R^p : y = f(x, u), x \in X(u)\} \quad (19)$$

The solution set $M(u, v)$ over the objective space $Y(u)$ and the decision space $D(v)$ is defined by

$$M(u, v) := \{x \in X(u) : f(x, u) \in N(u, v)\} \quad (20)$$

Sawaragi et al. [178] examine the stability of efficient solution sets when u varies and v is fixed. Kolbin and Perestoronin [115] study the stability of solutions to MOO problems based on normalization and the principle of solution selections. They consider the region of admissibility and scope of optimality using ε-stability.

D. Optimality vs. Complexity

One of the key issues in solving MOO problems is how to fine-tune the tradeoff between optimality (i.e., quality of optimal solutions) and complexity (i.e., efficiency in running time or space). We provide a summary of optimality and complexity in Table X for scalarization-based and metaheuristics MOO solution techniques discussed in Section IV.

Hybrid metaheuristics and trust-based MOO solutions also are heuristics as the variants of metaheuristics or combination with scalarization-based MOO methods to reduce solution search complexity. Thus, when considering hybrid approaches using either metaheuristics or trust, Table X provides a guideline towards what MOO methods to combine for both improved solution quality (i.e., optimality) and efficiency (i.e., complexity).

Quality of solutions in terms of Pareto optimality or near-optimal solutions is affected by the characteristics of the given problem. However, as far as a solution technique provides the same or close to optimal solutions, efficiency is the priority in selecting a solution technique. In particular, to solve large size problems, many variants of known MOO techniques have been introduced in order to reduce the complexity of a given algorithm. As discussed in Section IV-C, there have been efforts to combine more than one techniques for efficiency. Further, many metaheuristics approaches are proposed to generate higher solution diversity which may provide better solution quality with less overhead, such as reaching a global optimal solution with less convergence time.

Many MOO solution methods use constraints or a set of ideal solutions as reference or target points. Increasing the number of objectives makes a given MOO problem harder to solve as does increasing the number of constraints. In addition, it is highly challenging to identify the right set of weights, utility functions, or ideal objective values which critically impact the feasible solution space and complexity.

As shown in Table X, although there is a tradeoff between complexity and optimality (i.e., a highly expensive, slow process gives a better chance to find more optimal solutions while an efficient, fast process may not provide high quality solutions), the optimality achieved by a solution technique with computationally high complexity may not have good performance depending on context. Therefore, which MOO methods to choose still depends on the performance level required by a system or a decision maker.

VIII. Future Research Directions

An insight from the survey is that specific solution techniques for MOO problems (as discussed in Sections IV and VI) are often domain-specific and context-dependent (e.g., different context under 3 classes proposed in Section VI). We found that for Class 1 problems with a large search space, evolutionary algorithms are the most popular solution techniques. For Class 2 problems where two-party objectives exist with identical payoff functions, economic theories including game and auction theories are the dominant techniques. For Class 3 problems where two-party objectives exist with distinct individual payoff functions, cooperative game theory dominates. Trust-based methods follow the same trend and enhance these solution techniques as trust can be used as an integrated component in MOO problem formulation and system behavior modeling. In addition, trust-based heuristics can introduce high efficiency in solution search while achieving close-to-optimal solutions for a large size problem.

In many application domains of computing and networking, building trustworthy systems often can be a main concern where trust can be considered as an objective along with other objectives such as reliability, availability, security, or dependability. A promising potential research area is to develop MOO techniques to build trustworthy systems with multiple objectives where trust-based decisions can efficiently reduce solution search space. We suggest the following research directions:

- **Formulation of payoffs (or utilities):** As seen in MOO problems in Class 2 and Class 3, game theoretic approaches use the concept of payoffs and how to formulate each party's payoff significantly affects whether each party (or system) achieves its respective objective by obtaining expected payoff. Devising effective reward or penalty mechanisms to entice cooperative behaviors of individual entities to increase coalition and individual payoffs [44, 214] is a promising research direction.
- **Tradeoff analysis of objectives:** When multiple objectives exist in a system, what objective to prioritize, how to weigh each objective, or how to adjust constraints associated with achieving an objective function is closely related to the tradeoffs between multiple objectives. A future research direction is to devise a toolset allowing the end user to specify the MOO problem requirements, objectives and their relative priority, classify the MOO problem (see Section VI), select the best solution technique as suggested by the toolset (see Section IV), and more importantly, parameterize model parameters (give values to model parameters), and perform necessary tradeoff analyses to maximize the performance of applications or systems with multiple objectives.
- **Entity modeling:** An entity can be a machine, person, group, or organization. How to characterize each entity is closely related to how to measure performance of achieving system goals because an entity's performance or behavior affects system behavior. A promising research direction is to characterize an entity's features and behaviors through multiple QoS and social trust dimensions, including honesty, cooperativeness, social similarity, centrality, service quality, capability, etc. [41]. Based on game theoretic approaches, an individual's utility can be defined based on accrued trust so trust can play a role as a basis of decision making. Equally important is team modeling, i.e., how to measure the potential performance (synergy) of mixed teams, such as a team of machines and humans.
- **Attack modeling:** In various computing or networking environments, security goals are critical parts of system goals. How to model and characterize attack behaviors is closely related to how a system reaches a security standard. If attack behaviors are described unrealistically

TABLE X
OPTIMALITY AND COMPLEXITY OF MOO TECHNIQUES
($m = \#$FEASIBLE SOLUTIONS, $n = \#$OBJECTIVES, $k = \#$PARETO OPTIMAL SOLUTIONS, $r = \#$CONSTRAINTS $w =$CARDINALITY OF A WEIGHT SET)

Technique	Optimality	Complexity	Note	Ref.
Scalarization-based Methods				
Weighted sum	Weak/strong PO for convex objective functions	$O(mn)$	Difficult to choose weights based on preferences	[132, 131]
ε-constraints	Weak Pareto optimality (PO)	$O(nm^{n-1})$	Challenging to search for a set of constraints to identify tradeoff points among objectives; an adaptive algorithm exists with $O(k^{n-1})$	[132, 121]
Goal programming & Min-max	No guarantee for PO	$O(mn)$	Challenging to set a goal vector for identifying Pareto optimality	[54, 148]
Elastic constraints	No guarantee for PO; but	$O(nm(n-1))$	Difficult to set a penalty coefficient that significantly impacts solution search space; higher efficiency in solution search than ε-constraints	[69, 113]
Weighted metric	Weak PO	$O(wnm))$	Identifiable PO when a set of utopian objective values are known	[138, 140]
Achievement function	No guarantee for PO	$O((w+\rho)(n+m))$ where $\rho > 0$ which is an adjustable reference point	Incurs extra overhead to identify adaptive reference points; identifies better solution space	[138]
Benson's method	No guarantee for PO	$O(mnr)$	Increases feasible solution region; high impact of a random reference point on the existence of PO	[20, 56]
Utility (or value function)	No guarantee for PO	$O(mnr)$	Critical impact of utility functions on the existence of PO; Nash equilibrium solutions may not provide PO	[56, 85]
Metaheuristics				
Nondominated Sorting Genetic Algorithm (NSGA) & Strength Pareto Evolutionary Algorithm (SPEA)	No guarantee for PO; near-optimal solutions	$O(nm^3)$	NSGA II provides better complexity of $O(nm^2)$; useful for scalability and increasing the proximity to PO	[55]
Pareto Archived Evolutionary Strategy (PAES)	No guarantee for PO; near-optimal solutions	$O(amn)$ where a is the length of the archive	a is typically proportional to n, leading to complexity $O(mn^2)$; useful for a large size problem	[55]
Multiobjective Quantum-inspired Evolutionary Algorithm (MQEA)	No guarantee for PO; near-optimal solutions	$O(nq^c)$ where q is the number of qbits and $c > 0$	Significant improvement of the proximity to Pareto optimality but incurs high computational cost	[111, 173]
Hierarchical EA (HEA)	No guarantee for PO; near-optimal solutions	$O(n^c m)$ where $c > 0$	computationally efficient by removing valid or dominating solutions; increase of feasible solution space	[53, 61]
Ant Colony Optimization (ACO)	No guarantee for PO; near-optimal solutions	$O(nm \log m)$	Good for avoiding local optimal solutions; time-consuming process for convergence to the optimal solutions; $O(nm \log m)$ is for a fairly large evaporation factor	[24, 63, 151]
Particle Swarm Optimization (PSO)	No guarantee for PO; near-optimal solutions	$O(nm^2)$	No mathematical basis to identify optimal solutions; producing high quality solutions	[3, 109]
Simulated Annealing (SA)	No guarantee for PO; near-optimal solution	$O(n(m^2+m)\log m)$	Increasing feasible solution space	[92]
Tabu Search (TS)	No guarantee for PO; near-optimal solutions	$O(nm^2)$ per iteration	$O(nm \log m)$ per iteration with the robust TS; $O(nm)$ per iteration for a sized problem	[83, 161]
Variable Neighborhood Search (VNS)	No guarantee for PO; near-optimal solutions	nmr	Similar to Benson's method; but the size of m can be reduced by local search	[93, 146]

or the degree of attack intensity is considered incorrectly, optimizing system performance and/or security is not measured accurately. Modeling accurate, realistic attack behaviors is vital to measuring accurate optimization solutions. A promising future research direction is to model the dynamics between attackers and defenders by leveraging game theoretic approaches [2, 37, 144, 145] and reason how the system can dynamically adjust the parameters associated with a solution technique to best achieve multi-objective optimization.

- **Integration of trust into MOO problem formulation:** In using optimization techniques, a system or entity can use trust for decision making, which can be critical to identifying correct optimal solutions. For example, when using traditional evolutionary optimization techniques, the formulation of the fitness function is important to identifying optimal solution(s). If we use trust as a factor to calculate the value of fitness, the trust estimator and the accuracy of the estimated trust will affect the value of fitness and accordingly generate different optimal solutions. In addition, if we use a game theoretic approach to model a system where multiple parties want to share a total payoff, how each entity estimates trust towards another entity will affect how the payoff is shared. A promising future research area is to consider multidimensional trust comprising QoS and social trust components such as service quality, capability, honesty, cooperativeness, centrality, social contact, etc. [211, 212] and integrate trust into the optimization techniques for best achieving multi-objective optimization.

IX. SUMMARY

This paper provides a comprehensive survey of solution and modeling methods to solve multi-objective optimization (MOO) problems covering a wide range of applications/systems, including coalition formation (or team composition), task assignment, task scheduling, and resource allocation. By providing key advantages and disadvantages of each method, this work provides guidelines on how each method can be utilized in an application. We classified the papers published between 2000 and 2016 on MOO problems into three classes based on the design concept of global welfare vs. individual welfare.

As a future research direction, we suggest entity modeling, attack and defense behavior modeling, integration of multidimensional trust into multi-objective optimization problem formulation, payoff function modeling and formulation, and tradeoff analysis of multiple objectives, as future research directions with trust-based heuristics that can introduce high efficiency and effectiveness in solving MOO problems.

Acknowledgment

This work was supported in part by the U. S. Army Research Laboratory and the U. S. Army Research Office under contract number W911NF-12-1-0445. This research was also partially supported by the Department of Defense (DoD) through the office of the Assistant Secretary of Defense for Research and Engineering (ASD (R&E)). The views and opinions of the author(s) do not reflect those of the DoD or ASD (R&E).

References

[1] M. Agarwal, L. Vig, and N. Kumar, *Intelligent Robotics and Applications*, ser. Lecture Notes in Computer Science. Springer, 2011, vol. 7101, ch. Multi-objective robot coalition formation for non-additive environments.

[2] H. Al-Hamadi and I. Chen, "Adaptive network management for countering smart attack and selective capture in wireless sensor networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 3, pp. 451–466, 2015.

[3] B. S. Al-Kazemi and S. J. Habib, "Complexity analysis of problem-dimension using pso," in *Proceedings of the 7th WSEAS International Conference on Evolutionary Computing*. World Scientific and Engineering Academy and Society (WSEAS), 2006, pp. 45–52.

[4] O. Amin, E. Bedeer, M. H. Ahmed, and O. A. Dobre, "Energy efficiency and spectral efficiency trade-off for OFDM systems with imperfect channel estimation," in *2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 3553–3558.

[5] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Online team formation in social networks," in *Proc. 21st International Conference on World Wide Web*, Lyon, France, 2012, pp. 839–848.

[6] M. Analoui and M. Rezvani, "Toward microeconomic allocation of resources in multi-service overlay networks," *Journal of Computer and Systems Sciences International*, vol. 50, no. 5, pp. 741–755, 2011.

[7] E. Asl, J. Bentahar, H. Otrok, and R. Mizouni, "Efficient community formation for web services," *IEEE Transactions on Services Computing*, vol. 8, no. 4, pp. 586–600, July 2015.

[8] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer, Dec. 2002.

[9] L. M. Ausubel and P. Milgrom, *Cominatorial Auctions*. MIT Press, 2006, ch. The lovely but lonely Vickrey auction.

[10] H. Aziz and F. Brandl, "Existence of stability in hedonic coalition formation games," in *Proceedings of 11th International Conference on Autonomous Agents and Multiagent Systems*, 2012, pp. 763–770.

[11] J. Balicki, "An adaptive quantum-based multiobjective evolutionary algorithm for efficient task assignment in distributed systems," in *Proc. World Scientific and Engineering Academy and Society 13th International Conference on Computers*, 2009, pp. 417–422.

[12] S. Banerjee, H. Konishi, and T. Sonmez, "Core in a simple coalition formation game," *Social Choice and Welfare*, vol. 18, no. 1, pp. 135–153, 2001.

[13] R. Baños, C. Gil, B. Paechter, and J. Ortega, "A hybrid meta-heuristic for multi-objective optimization: Mosats," *Journal of Mathematical Modelling and Algorithms*, vol. 6, no. 2, pp. 213–230, 2007.

[14] P. Beautement, "Agile and adaptive coalition operations - leveraging the power of complex environments," in *11th International Command and Control Research and Technology Symposium (ICCRTS)*, San Diego, CA, 20-22 June 2006.

[15] E. Bedeer, O. A. Dobre, M. H. Ahmed, and K. E. Baddour, "Joint optimization of bit and power loading for multicarrier systems," *IEEE Wireless Communications Letters*, vol. 2, no. 4, pp. 447–450, August 2013.

[16] ——, "A multiobjective optimization approach for optimal link adaptation of OFDM-based cognitive radio systems with imperfect spectrum sensing," *IEEE Transactions on Wireless Communications*, vol. 13, no. 4, pp. 2339–2351, April 2014.

[17] ——, "Rate-interference tradeoff in OFDM-based cognitive radio systems," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4292–4298, Sept 2015.

[18] ——, "A systematic approach to jointly optimize rate and power consumption for OFDM systems," *IEEE Transactions on Mobile Computing*, vol. 15, no. 6, pp. 1305–1317, June 2016.

[19] A. Bedoui, P. Debreux, L. Idoumghar, R. Schott, and T. Schott, "Steps toward the design of hybrid metaheuristics for the multi-objective frequency assignment problem in broadcasting," in *2014 IEEE 2nd International Symposium on Telecommunication Technologies (ISTT)*, Nov 2014, pp. 208–213.

[20] H. Benson, "Existence of efficient solutions for vector maximization problems," *Journal of Optimization Theory and Applications*, vol. 26, no. 4, pp. 569–580, 1978.

[21] D. Bertisimas and J. Tsitsiklis, "Simulated annealing," *Statistical Science*, vol. 8, no. 1, pp. 10–15, 1993.

[22] J. T. Blake and M. W. Carter, "A goal programming approach to strategic resource allocation in acute care hospitals," *European Journal of Operational Research*, vol. 140, no. 3, pp. 541–561, 2002.

[23] F. Bloch and E. Diamantoudi, "Noncooperative formation of coalitions in hedonic games," *International Journal of Game Theory*, vol. 40, no. 2, pp. 264–280, 2011.

[24] C. Blum, "Ant colony optimization: Introduction and recent trends," *Physics of Life Reviews*, vol. 2, no. 4,

2005.

[25] ——, "Revisiting dynamic programming for finding optimal subtrees in trees," *European Journal of Operational Research*, vol. 177, no. 1, pp. 102–115, 2007.

[26] ——, *Hybrid Metaheuristics in Combinatorial Optimization: A Tutorial*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 1–10.

[27] L. Blumrosen and N. Nisan, *Algorithmic Game Theory*. Campridge University Press, 2007, ch. Combinatorial auctions, pp. 267–298.

[28] A. Bogomolnaia and M. Jackson, "The stability of hedonic coalition structures," *Games and Economic Behavior*, vol. 38, no. 2, pp. 201–230, 2002.

[29] A. Brandenburger, "Cooperative game theory: Characteristic functions, allocations, marginal contribution," Stern School of Business, New York University, Tech. Rep., 2007.

[30] S. Breban and J. Vassileva, "Using inter-agent trust relationships for efficient coalition formation," in *Proc. 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence*, Calgary, Canada, May 2002, pp. 221–236.

[31] M. Brown, B. An, C. Kiekintveld, F. Ordóñez, and M. Tambe, "An extended study on multi-objective security games," *Autonomous Agents and Multi-Agent Systems*, vol. 28, no. 1, pp. 31–71, 2014.

[32] Y. Cai, D. Wu, and L. Zhou, "Green resource sharing for mobile device-to-device communications," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2014, pp. 3142–3147.

[33] Y. Censor, "Pareto optimality in multiobjective optimization," *Applied Mathematics and Optimization*, vol. 4, no. 1, pp. 41–59, 1977.

[34] V. Chandrasekhar, J. G. Andrews, T. Muharemovic, Z. Shen, and A. Gatherer, "Power control in two-tier femtocell networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 8, pp. 4316–4328, Aug. 2009.

[35] K.-H. Chang, *Design Theory and Methods using CAD/CAE: The Computer Aided Engineering Design Series*. Academic Press, Technology & Engineering, 2014.

[36] M. Chang, J. Cho, I. Chen, K. Chan, and A. Swami, "Trust-based task assignment in military tactical networks," in *17th International Command and Control Research and Technology Symposium (ICCRTS)*, Fairfax, Virginia, June 2012.

[37] I. Chen, R. Mitchell, and J. Cho, "On modeling of adversary behavior and defense for survivability of military manet applications," in *34th IEEE Military Communications Conference(MILCOM 2015)*, Tampa, FL, Oct. 2015.

[38] J. Chen and D. Sun, "Resource constrained multi-robot task allocation based on leaderfollower coalition methodology," *The International Journal of Robotics Research*, vol. 30, no. 12, pp. 1423–1434, 2011.

[39] L. Cherchye, T. Demuynck, and B. de Rock, "Is utility transferable? a revealed preference analysis," *Theoretical Economics*, vol. 10, pp. 51–65, 2015.

[40] J. H. Cho, I. R. Chen, Y. Wang, and K. S. Chan, "Trust-based multi-objective optimization for node-to-task assignment in coalition networks," in *2013 International Conference on Parallel and Distributed Systems*, Dec 2013, pp. 372–379.

[41] J. Cho, A. Swami, and I. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 4, pp. 562–583, Nov. 2011.

[42] J. Cho, A. Swami, and T. Cook, "Combinatorial auction-based multiple dynamic mission assignment," in *IEEE Military Communications Conference*, Nov. 2011, pp. 1327–1332.

[43] J. Cho, I. Chen, Y. Wang, K. Chan, and A. Swami, "Multi-objective optimization for trustworthy tactical networks: A survey and insights," in *18th International Command and Control Research and Technology Symposium (ICCRTS)*, Alexandria, Virginia, June 2013.

[44] J.-H. Cho, "Tradeoffs between trust and survivability for mission effectiveness in tactical networks," *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 754–766, 2015.

[45] E. R. Clayton, W. E. Weber, and B. W. T. III, "A goal programming approach to the optimization of multi response simulation models," *A I I E Transactions*, vol. 14, no. 4, pp. 282–287, 1982.

[46] C. A. C. Coello, "An updated survey of GA-based multiobjective optimization techniques," *ACM Computing Surveys*, vol. 32, no. 2, pp. 109–143, 2000.

[47] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. Springer, 2007.

[48] C. A. C. Coello, "Evolutionary multi-objective optimization: a historical view of the field," *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28–36, 2006.

[49] V. Dang, R. Dash, A. Rogers, and N. Jennings, "Overlapping coalition formation for efficient data fusion in multi-sensor networks," in *AAAI*, vol. 6, 2006, pp. 635–640.

[50] A. Das and M. M. Islam, "SecuredTrust: a dynamic trust computation model for secured communication in multiagent systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 2, pp. 261–274, March/April 2012.

[51] B. Das, S. Misra, and U. Roy, "Coalition formation for cooperative service-based message sharing in vehicular ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 144–156, Jan. 2016.

[52] D. Dasgupta, D. Garrett, F. Nino, A. Banceanu, and D. Becerra, *A Genetic-Based Solution to the Task-Based Sailor Assignment Problem*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 167–203.

[53] E. D. de Jong, D. Thierens, and R. A. Watson, *Hierarchical Genetic Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 232–241.

[54] K. Deb, "Solving goal programming problems using multi-objective genetic algorithms," in *Proceedings of 1999 Congress on Evolutionary Computation*, 1999.

[55] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, April 2002.

[56] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc, 2001.

[57] S. Deng, "Characterizations of the nonemptiness and compactness of solution sets in convex vector optimization," *Journal of Optimization Theory and Applications*, vol. 96, no. 1, pp. 201–209, 1998.

[58] ——, "On efficient solutions in vector optimization," *Journal of Optimization Theory and Applications*, vol. 96, no. 1, pp. 201–209, 1998.

[59] U. Deshpande, A. Gupta, and A. Basu, "Coordinated problem solving through resource sharing in a distributed environment," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, no. 2, pp. 1299–1304, April 2004.

[60] O. Dib, M. A. Manier, and A. Caminada, "A hybrid metaheuristic for routing in road networks," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 765–770.

[61] B. Dieber, C. Micheloni, and B. Rinner, "Resource-aware coverage and task assignment in visual sensor networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1424–1437, Oct. 2011.

[62] Y. Donoso and R. Fabregat, *Multi-Objective Optimization in Computer Networks Using Metaheuristics*. Auerbach Publications, Taylors and Francis Group, 2007.

[63] M. Dorigo and T. Sttzle, *Ant Colony Optimization*. MIT, 2004.

[64] C. Dorn, F. Skopik, D. Schall, and S. Dustdar, "Interaction mining and skill-dependent recommendations for multi-objective team composition," *Data Knowledge Engineering*, vol. 70, no. 10, pp. 866–891, Oct. 2011.

[65] O. Dridi, S. Krichen, and A. Guitouni, "A multi-objective optimization approach for resource assignment and task scheduling problem: application to maritime domain," in *Proceedings of IEEE World Congress on Computational Intelligence*, Brisbane, Australia, June 2012.

[66] R. Duan, R. Prodan, and X. Li, "Multi-objective game theoretic schedulingof bag-of-tasks workflows on hybrid clouds," *IEEE Transactions on Cloud Computing*, vol. 2, no. 1, pp. 29–42, Jan. 2014.

[67] L. Duckstein, *Multiobjective optimization in structural design: The model choice problem*. New York, NY: John Wiley and Sons, Inc., 1984, pp. 459–481.

[68] N. Edalat, C. Than, and W. Xiao, "An auction-based strategy for distributed task allocation in wireless sensor networks," *Computer Communications*, vol. 35, no. 8, pp. 916–928, 2012.

[69] M. Ehrgott, "A discussion of scalarization techniques for multiple objective integer programming," *Annals of Operations Research*, vol. 147, no. 1, pp. 343–360, 2006.

[70] M. Ehrgott and D. M. Ryan, *The Method of Elastic Constraints for Multiobjective Combinatorial Optimization and its Application in Airline Crew Scheduling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, ch. Multi-Objective Programming and Goal Programming: Theory and Applications, pp. 117–122.

[71] T. S. Ferguson, *Game Theory*, 2nd ed. Mathematics Department, UCLA, 2014, ch. Games in Coalitional Form.

[72] J. E. Fieldsend and R. M. Everson, "Multi-objective optimisation in the presence of uncertainty," in *2005 IEEE Congress on Evolutionary Computation*, vol. 1, Sept 2005, pp. 243–250.

[73] P. J. Fleming and A. P. Pashkevich, "Computer aided control system design using a multiobjective optimization approach," *Control*, vol. 85, pp. 174–179, 1985.

[74] K. Fleszar at al., "The complexity of solving multiobjective optimization problems and its relation to multivalued functions," *Electronic Colloquium on Computational Complexity*, 2011.

[75] Q. Gao, J. Zhang, X. Shen, and B. Larish, "A cross-layer optimization approach for energy efficient wireless sensor networks: coalition-aided data aggregation, cooperative communication, and energy balancing," *Advances in Multimedia*, vol. 2007, no. 1, Jan. 2007.

[76] M. R. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman & Co., 1979.

[77] V. K. Garg, Y. Narahari, and M. Murty, "Novel biojective clustering (BiGC) based on cooperative game theory," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 5, pp. 1070–1082, May 2013.

[78] T. Genin and S. Aknine, "Coalition formation strategies for self-interested agents in task oriented domains," in *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2010, pp. 205–212.

[79] O. N. Gharehshiran, A. Attar, and V. Krishnamurthy, "Collaborative sub-channel allocation in cognitive lte femto-cells: a cooperative game-theoretic approach," *IEEE Transactions on Communications*, vol. 61, no. 1, pp. 325–334, Jan. 2013.

[80] C. Glaber et al., "Hardness and approximability in multi-objective optimization," *Electronic Colloquium on Computational Complexity*, no. 31, 2010.

[81] F. Glover, "Tabu search part i," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

[82] ——, "Tabu search part ii," *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4–32, 1990.

[83] F. Glover and M. Laguna, *Tabu Search*. Boston: Kluwer Academic Publishers, 1997.

[84] T. Goel and N. Stander, "Multi-objective optimization using LS-OPT," in *LS-DYNA Forum*, Frankenthal, Germany, 2007.

[85] S. Greco, K. Klamroth, J. D. Knowles, and G. Rudolph, "Understanding complexity in multiobjective optimization," Dagstuhl Seminar 15031, Tech. Rep. 1, 2015, dagstuhl Reports.

[86] M. Guazzone, C. Anglano, and M. Sereno, "A game-theoretic approach to coalition formation in green cloud federations," in *14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2014),*, May 2014, pp. 618–625.

[87] L. Guo, X. Wang, and G. Zeng, "Using trust and self-confidence for coalition formation in migrating workflow," in *International Forum on Information Technology and Applications*, 2009, pp. 375–378.

[88] L. Guo, G. Shao, and S. Zhao, "Multi-objective task assignment in cloud computing by particle swarm optimization," in *8th International Conference on Wireless Communications, Networking and Mobile Computing*, Shanghai, China, Sept. 2012.

[89] Y. Guo, J. Ma, C. Wang, and K. Yang, "Incentive-based optimal nodes selection mechanism for threshold key management in manets with selfish nodes," *International Journal of Distributed Sensor Networks*, vol. 2013, no. Article ID 416983, 2013.

[90] V. Ha and L. Le, "Distributed base station association and power control for heterogeneous cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 282–296, Jan. 2014.

[91] S. Hajri-Gabouj, "A fuzzy genetic multiobjective optimization algorithm for a multilevel generalized assignment problem," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 33, no. 2, pp. 214–224, May 2003.

[92] P. B. Hansen, "Simulated annealing," Syracuse University, School of Computer and Information Science, Tech. Rep., 1992, paper 170.

[93] P. Hansen and N. Mladenović, *Variable Neighborhood Search*. Boston, MA: Springer US, 2014, pp. 313–337.

[94] N. Hazon, Y. Aumann, and S. Kraus, "Collaborative multi agent physical search with probabilistic knowledge," in *Proceedings of 21st International Joint Conference on Artificial Intelligence*, Pasadena, CA, July 2009.

[95] S. Helbig, "Approximation of the efficient point set by perturbation of the ordering cone," *ZOR - Methods and Models of Operations Research*, vol. 35, no. 3, pp. 197–220, 1991.

[96] J.-J. Huang, C.-Y. Chen, H.-H. Liu, and G.-H. Tzeng, "A multiobjective programming model for partner selection-perspectives of objective synergies and resource allocation," *Expert Systems with Applications*, vol. 37, no. 5, pp. 3530–3536, May 2010.

[97] H. Huo, L. Yue, and X. Shen, "Trust game analysis between partners in the supply chain alliance," in *International Conference on Management Science and Industrial Engineering*, 2011, pp. 947–950.

[98] C.-L. Hwang and A. S. M. Masud, *MultipleObjective Decision Making - Methods and Applications*. Berlin, Germany: Springer, 1979.

[99] M. Hyodo, T. Matsuo, and T. Ito, "An optimal coalition formation among buyer agents based on a genetic algorithm," in *Developments in Applied Artificial Intelligence*. Springer, 2003, pp. 759–767.

[100] D. A. Iancu and N. Trichakis, "Pareto efficiency in robust optimization," *Management Science*, vol. 60, no. 1, pp. 130–147, 2014.

[101] C. Jiang, Y. Shi, S. Kompella, Y. T. Hou, and S. F. Midkiff, "Bicriteria optimization in multihop wireless networks: Characterizing the throughput-energy envelope," *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1866–1878, Sept 2013.

[102] Y. Jiang and J. Jiang, "Contextual resource negotiation-based task allocation and load balancing in complex software systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 5, pp. 641–653, May 2009.

[103] L. Jiao, J. Lit, W. Du, and X. Fu, "Multi-objective data placement for multi-cloud socially aware services," in *Proceedings of INFOCOM*. IEEE, 2014, pp. 28–36.

[104] Y. Jin, J. Jin, A. Gluhak, K. Moessaner, and M. Palaniswami, "An intelligent task allocation scheme for multihop wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 444–451, March 2012.

[105] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments - a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.

[106] M. D. Johnston, "Deep space network scheduling using multi-objective optimization with uncertainty," in *SpaceOps 2008 Conference*, Heidelberg, Germany, May 2008.

[107] D. Jones and M. Tamiz, *International Series in Operations Research and Management Science*, 2010, vol. 141, ch. Practical Goal Programming.

[108] S. Kartik and C. Murthy, "Task allocation algorithms for maximizing reliability of distributed computing systems," *IEEE Transactions on Computers*, vol. 46, no. 6, pp. 719–724, 1997.

[109] J. Kennedy, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[110] Z. Khan, J. Lehtomaki, M. C. amd M. Latva-aho, and L. A. DaSilva, "Throughput-efficient dynamic coalition formation in distributed cognitive radio networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, no. 87, April 2010.

[111] Y. Kim, J.-H. Kim, and K.-H. Han, "Quantum-inspired multiobjective evolutionary algorithm for multiobjective 0/1 knapsack problems," in *IEEE Congress on Evolutionary Computation*, Vancouver, Canada, 16-21 July 2006.

[112] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[113] K. Klamroth and T. Jørgen, "Constrained optimization using multiple objective programming," *Journal of Global Optimization*, vol. 37, no. 3, pp. 325–355, March

2007.

[114] J. D. Knowles and D. W. Corne, “The pareto archive evolution strategy: A new baseline algorithm for multi-objective optimization,” in *Proceedings of 1999 Congress on Evolutionary Computation*, 1999, pp. 98–105.

[115] V. V. Kolbin and D. S. Perestoronin, “Several problems of the stability of multi-objective optimization,” in *International Conference “Stability and Control Processes” in Memory of V.I. Zubov (SCP)*, Oct 2015, pp. 16–19.

[116] G. Koloniari and E. Pitoura, “A game-theoretic approach to the formation of clustered overlay networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, pp. 589–59, April 2012.

[117] A. Konaka, D. W. Coit, and A. E. Smith, “Multi-objective optimization using genetic algorithms: A tutorial,” *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.

[118] V. Krishna, *Auction Theory*, 2nd ed. Elsevier, 2010.

[119] H. W. Kuhn and A. W. Tucker, “Nonlinear programming,” in *Proceedings of 2nd Berkeley Symposium on Mathematical Statistics and Probability*, University of California at Berkeley, Berkeley, CA, 1951, pp. 481–492.

[120] S. Kukkonen and K. Deb, *A Fast and Effective Method for Pruning of Non-dominated Solutions in Many-Objective Problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 553–562.

[121] M. Laumanns, L. Thiele, and E. Zitzler, “An adaptive scheme to generate the pareto front based on the epsilon-constraint method,” in *Practical Approaches to Multi-Objective Optimization*, ser. Dagstuhl Seminar Proceedings, J. Branke, K. Deb, K. Miettinen, and R. E. Steuer, Eds., no. 04461. Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), 2005.

[122] C.-H. Lee and K. Shin, “Optimal task assignment in homogeneous networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 8, no. 2, pp. 119–129, 1997.

[123] K. Y. Lee and J. B. Park, “Application of particle swarm optimization to economic dispatch problem: advantages and disadvantages,” in *IEEE Power Systems Conference and Exposition*, 2006, pp. 188–192.

[124] C. Li and K. Sycara, “Algorithm for combinatorial coalition formation and payoff division in an electronic marketplace,” in *Proceedings of 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, 2001, pp. 120–127.

[125] W. Li, F. C. Delicato, and A. Y. Zomaya, “Adaptive energy-efficient scheduling for hierarchical wireless sensor networks,” *ACM Transaction of Sensor Networks*, vol. 9, no. 3, pp. 33:1–33:34, May 2013.

[126] Z. Li, Z. Li, and G. Rudolph, *Advanced Intelligent Computing Theories and Applications: With Aspects of Contemporary Intelligent Computing Techniques Communications in Computer and Information Science*, 2007, vol. 2, ch. On the convergence properties of quantum-inspired multi-objective evolutionary algorithms, pp. 245–255.

[127] Q. Liao, S. Staczak, and F. Penna, “A statistical algorithm for multi-objective handover optimization under uncertainties,” in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2013, pp. 1552–1557.

[128] P. Limbourg, *Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer, 2005, vol. 3410, ch. Multi-objective Optimization of Problems with Epistemic Uncertainty.

[129] L. Lin and J. Huai, “QGrid: an adaptive trust aware resource management framework,” *IEEE Systems Journal*, vol. 3, no. 1, pp. 78–90, March 2009.

[130] W.-Y. Liu, K. Yue, T.-Y. Wu, and M.-J. Wei, “An approach for multi-objective categorization based on the game theory and markov process,” *Applied Soft Computing*, vol. 11, no. 6, pp. 4087–4096, Set. 2011.

[131] Z. Q. Luo and S. Zhang, “Dynamic spectrum management: Complexity and duality,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 57–73, Feb 2008.

[132] R. T. Marler and J. S. Arora, “Survey of multi-objective optimization methods for engineering,” *Structure Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, April 2004.

[133] L. Mashayekhy and D. Grosu, “A merge-and-split mechanism for dynamic virtual organization formation in grids,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 540–549, Mar. 2014.

[134] N. F. Matsatsinis and P. Delias, “AgentAllocator: An agent-based multi-criteria decision support system for task allocation,” in *1st International Conference on Applications of Holonic and Multi-Agent Systems, Prague, Czech Republic, Lecture Notes in Computer Science*, vol. 2744, Sept. 2003, pp. 1082–1083.

[135] R. Meng, Y. Ye, and N. G. Xie, “Multi-objective optimization design methods based on game theory,” in *8th World Congress on Intelligent Control and Automation*, 2010, pp. 2220–2227.

[136] Merriam and Webster Dictionary, 2015.

[137] M. Mezmaz, N. Melab, and E. G. Talbi, “A parallel exact hybrid approach for solving multi-objective problems on the computational grid,” in *Proceedings 20th IEEE International Parallel Distributed Processing Symposium*, April 2006.

[138] K. Miettinen, *Multiobjective Optimization*, ser. LNCS 5252. Springer-Verlag Berlin Heidelberg, 2008, ch. Introduction to Multiobjective Optimization: Noninteractive Approaches, pp. 1–26.

[139] ——, *Nonlinear Multiobjective Optimization*, ser. International Series in Operations Research & Management Science. Springer Science + Business Media, LLC, 1998, vol. 12.

[140] ——, *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999, vol. 12, international Series in Operations Research & Management Science.

[141] D. G. Mikulski, F. L. Lewis, E. Y. Gu, and G. R. Hudas,

"Trust dynamics in multi-agent coalition formation," in *Proceedings of SPIE, Unmanned Systems Technology XIII*, vol. 8045, April 2011.
[142] F. Milano, C. A. Canizares, and M. Invernizzi, "Multiobjective optimization for pricing system security in electricity markets," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 596–604, May 2003.
[143] L. Militano, A. Lera, and F. Scarcello, "A fair cooperative content-sharing service," *Computer Networks*, vol. 57, no. 9, pp. 1955–1973, June 2013.
[144] R. Mitchell and I. Chen, "Effect of intrusion detection and response on reliability of cyber physical systems," *IEEE Transactions on Reliability*, vol. 62, no. 1, pp. 199–210, March 2013.
[145] ——, "Modeling and analysis of attacks and counter defense mechanisms for cyber physical systems," *IEEE Transactions on Reliability*, 2015, in pess.
[146] N. Mladenovi/'c and P. Hansen, "Variable neighborhood search," *Computers and Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997.
[147] C. Murthy, "Optimal task allocation in distributed systems by graph matching and state space search," *Journal of Systems and Software*, vol. 46, no. 1, pp. 59–75, 1999.
[148] H. Nakayama and Y. Yun, "Combining aspiration level methods in multi-objective programming and sequential approximate optimization using computational intelligence," in *IEEE Symposium on Computational Intelligence in Multicriteria Decision Making*, 2007, pp. 319–324.
[149] H. Nakayama, *Duality in Multi-Objective Optimization*. Boston, MA: Springer US, 1999, pp. 69–97.
[150] L. Nardin and J. Sichman, "Simulating the impact of trust in coalition formation: a preliminary analysis," in *2nd Brazilian Workshop on Social Simulation*, 2010, pp. 33–40.
[151] F. Neumann, D. Sudholt, and C. Witt, *Computational Complexity of Ant Colony Optimization and Its Hybridization with Local Search*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 91–120.
[152] N. Nisan, T. Roughgarden, E. Targos, and V. V. Vazirani, *Algorithmic Game Theory*. Cambridge University Press, Sept. 2007.
[153] L. Nogueira and L. M. Pinho, "Time-bounded distributed QoS-aware service configuration in heterogeneous cooperative environments," *Journal of Parallel and Distributed Computing*, vol. 69, no. 4, pp. 491–507, June 2009.
[154] C. Notario, R. Baert, and M. DHondt, "Multi-objective genetic algorithm for task assignment on heterogeneous nodes," *International Journal of Digital Multimedia Broadcasting*, pp. 1–12, 2012, article ID 716780.
[155] J. Oh, *Cooperative game theory with nontransferable utility*. Wiley Encyclopedia of Operations Research and Management Science, 2011.
[156] T. Okahe, Y. Jin, and B. Sendhoff, "A critical survey of performance indices for multi-objective optimisation," in *The 2003 Congress on Evolutionary Computation (CEC'2003)*, vol. 2, 2003.
[157] A. Osyczka, "An approach to multicriterion optimization problems for engineering design," *Computer Methods in Applied Mechanics and Engineering*, vol. 15, no. 3, pp. 309–333, Sept. 1978.
[158] A. Ouni, H. Rivano, F. Valois, and C. Rosenberg, "Energy and throughput optimization of wireless mesh networks with continuous power control," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 1131–1142, Feb. 2015.
[159] V. Pareto, *The rise and fall of the elites*. Bedminister Press, 1968.
[160] D. C. Parkes, "Iterative combinatorial auctions: achieving economic and computational efficiency," Ph.D. dissertation, Computer and Information Science, University of Pennsylvania, 2001.
[161] G. Paul, "An efficient implementation of the robust tabu search heuristic for sparse quadratic assignment problems."
[162] G. Pesant and M. Gendreau, "A constraint programming framework for local search methods," *Journal of Heuristics*, vol. 5, pp. 155–179, 1999.
[163] P. S. Pillai and S. Rao, "Resource allocation in cloud computing using the uncertainty principle of game theory," *IEEE Systems Journal*, vol. 10, no. 2, pp. 637–648, June 2016.
[164] D. Pisinger and S. Ropke, *Handbook of Metaheuristics*, ser. International Series in Operations Research & Management Science. Berlin: Springer, 2010, vol. 146, ch. Large neighborhood search, pp. 399–419.
[165] A. Rachedi and A. Benslimane, "Multi-objective optimization for security and QoS adaptation in wireless sensor networks," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–7.
[166] G. R. Raidl, *A Unified View on Hybrid Metaheuristics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–12.
[167] S. Ramchurn, M. Polukarov, A. Farinelli, C. Truong, and N. Jennings, "Coalition formation with spatial and temporal constraints," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, vol. 3. International Foundation for Autonomous Agents and Multiagent Systems, 2010, pp. 1181–1188.
[168] G. P. Rangaiah, *Multi-Objective Optimization: Techniques and Applications in Chemical Engineering*. World Scientific Publishing, 2009.
[169] S. S. Rao, "Game theory approach for multi-objective structural optimization," *Computers and Structures*, vol. 25, no. 1, pp. 119–127, 1987.
[170] M. Reyes-Sierra and C. A. C. Coello, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 287–308, 2006.
[171] B. J. Ritzel, J. W. Eheart, and S. Ranjithan, "Using genetic algorithms to solve a multiple objective groundwater pollution containment problem," *Water Resources Research*, vol. 30, no. 5, pp. 1589–1603, 1994.
[172] C. Romero, *Handbook of critical issues in goal pro-*

gramming. Pergamon Press, Oxford, 1991.

[173] S.-J. Ryu, K.-B. Lee, B.-S. Yoo, T.-J. Kim, S.-J. Lee, and J.-H. Kim, *Multiobjective Quantum-Inspired Evolutionary Algorithm with Preference-Based Selection 2: Comparison Study*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 291–299.

[174] W. Saad, Z. Han, M. Debbah, A. Hjørungnes, and T. Başar, "Coalitional game theory for communication networks," *IEEE Signal Processing Magazine*, vol. 26, no. 5, pp. 77–97, 2009.

[175] W. Saad, T. Alpcan, T. Başar, and A. Hjørungnes, "Coalitional game theory for security risk management," in *IEEE 5th International Conference on Internet Monitoring and Protection*, May 2010, pp. 35–40.

[176] W. Saad, Z. Han, T. Başar, M. Debbah, and A. Hjørungnes, "Hedonic coalition formation for distributed task allocation among wireless agents," *IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1327–1344, Sept. 2011.

[177] S. Sariel-Talay, T. Balch, and N. Erdogan, "A generic framework for distributed multirobot cooperation," *Journal of Intelligent and Robotic Systems*, vol. 63, no. 2, pp. 323–358, Aug. 2011.

[178] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multiobjective Optimization*, ser. Mathematics in Science and Engineering, Richard and Bellman, Eds. London, UK: Academic Press, Inc., 1985, vol. 176.

[179] L. Shapley, "A value for n-person games," *Annals of Mathematical Studies*, vol. 28, 1953.

[180] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artificial Intelligence*, vol. 101, no. 1, pp. 165–200, 1998.

[181] L. Shen, L. Zhang, and D. Huang, "Trust-driven both-matched algorithm for grid task multi-objective scheduling," in *2nd International Conference on Information Science and Engineering*, 2010, pp. 1661–1664.

[182] C.-D. Shi and D.-X. Bian, "On partner selection and task assignment of logistic alliance under lean philosophy," in *Proceedings of IEEE International Conference on Grey Systems and Intelligent Services*, Nanjing, China, Nov. 2009.

[183] Q. Shi, C. Comaniciu, and K. Jaffres-Runser, "An auction-based mechanism for cooperative sensing in cognitive networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 3649–3661, 2013.

[184] Z. Shi, Q. Chen, S. Li, H. Cai, and X. Shen, "Cooperative task allocation for multiple mobile robots based on multi-objective optimization method," in *Proceedings of 3rd IEEE International Conference on Computer Science and Information Technology*, Chengdu China, July 2010.

[185] W. S. Shin and A. Ravindran, "Interactive multiple objective optimization: Survey I - continuous case," *Computers & Operations Research*, vol. 18, no. 1, pp. 97–114, 1991.

[186] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008, ch. Introduction to Noncooperative Game Theory: Games in Normal Form.

[187] S. Shroff and V. Dabhi, "Dew point modeling using GEP based multiobjective optimization," Computing Research Repository (CoRR), Tech. Rep., 2013.

[188] D. Shuai, Q. Shuai, and Y. Dong, "Particle model to optimize resource allocation and task assignment," *Information Systems*, vol. 32, no. 7, pp. 987–995, 2007.

[189] K. B. Sim, D. W. Lee, and J. Y. Kim, "Game theory based co-evolutionary algorithm: A new computational," *International Journal of Control, Automation, and Systems*, vol. 2, no. 4, pp. 467–474, 2004.

[190] C. Singh, S. Sarkar, and A. Aram, "Provider-customer coalitional games," *IEEE/ACM Transactions on Networking*, vol. 19, no. 5, pp. 1528–1542, 2011.

[191] L. Song, D. Niyato, Z. Han, and E. Hossain, "Game-theoretic resource allocation methods for device-to-device communication," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 136–144, June 2014.

[192] N. Srinivas and K. Deb, "Multi-objective optimizaiton using nondominated sorting in genetic algorithms," *Journal of Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, Fall 1994.

[193] S. Srinivasan and N. Jha, "Safety and reliability driven task allocation in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 10, no. 3, pp. 238–251, 1999.

[194] R. Subrata, A. Y. Zomaya, and B. Landfeldt, "A cooperative game framework for QoS guided job allocation schemes in grids," *IEEE Transactions on Computers*, vol. 57, no. 10, pp. 1413–1422, Oct. 2008.

[195] C. Szabo, Q. Z. Sheng, T. Kroeger, Y. Zhang, and J. Yu, "Science in the cloud: Allocation and execution of data-intensive scientific workflows," *Journal of Grid Computing*, vol. 12, no. 2, pp. 245–264, 2014.

[196] M. Tamiz and D. F. Jones, "A review of goal programming and its applications," *Annals of Operations Research*, vol. 58, pp. 39–53, 1995.

[197] M. Tamiz and D. J. S.K. Mirrazavi, "Extensions of pareto efficiency analysis to integer goal programming," *Omega*, vol. 27, pp. 179–188, 1999.

[198] K. Tan, T. Lee, and E. Khor, "Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons," *Artificial Intelligence Review*, vol. 17, no. 4, pp. 251–290, June 2002.

[199] A. Tino, "Multi-objective Tabu search based topology synthesis for designing power and performance efficient NoC architectures," Master's thesis, Ryerson University, Toronto, Ontario, Canada, 2011.

[200] A. T. Tolmidis and L. Petrou, "Multi-objective optimization for dynamic task allocation in a multi-robot system," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 5–6, pp. 1458–1468, May/June 2013.

[201] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, Jan. 1985.

[202] E. L. Ulungu and J. Teghem, "Multi-objective combinatorial optimization problems: A survey," *Journal of*

Multi-Criteria Decision Analysis, vol. 3, no. 2, pp. 83–104, 1994.

[203] P. J. M. van Laarhoven and E. H. L. Aarts, “Simulated annealing: theory and applications,” *Mathematics and Its Applications*, vol. 37, pp. 7–15, 1987.

[204] M. Vasquez and Y. Vimont, “Improved results on the 01 multidimensional knapsack problem,” *European Journal of Operational Research*, vol. 165, no. 1, pp. 70–81, 2005.

[205] D. A. V. Veldhuizen and G. B. Lamont, “Multiobjective evolutionary algorithms: analyzing the state-of-the-art,” *Evolutionary Computation*, vol. 8, no. 2, pp. 125–147, 2000.

[206] D. Vidyarthi and L. Khanbary, “Multi-objective optimization for channel allocation in mobile computing using NSGA-II,” *International Journal of Network Management*, vol. 21, pp. 247–266, May 2011.

[207] F. Wang, C. Xu, L. Song, and Z. Han, “Energy-efficient resource allocation for device-to-device underlay communication,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 4, pp. 2082–2092, April 2015.

[208] H. Wang, C. Li, C. Yan, Q. Li, and J. Li, “Ad hoc grid task scheduling algorithm considering trust-demand,” in *2nd International Conference on Future Computer and Communication*, vol. 3, Wuhan, China, 2010, pp. 109–113.

[209] H. Wang, L. Wang, and C. Yu, “Integrating trust with qualitative and quantitative preference for service selection,” in *2014 IEEE International Conference on Services Computing*, June 2014, pp. 299–306.

[210] T. Wang, L. Song, Z. Han, and W. Saad, “Distributed cooperative sensing in cognitive radio networks: An overlapping coalition formation approach,” *IEEE Transactions on Communications*, vol. 62, no. 9, pp. 3144–3160, Sept. 2014.

[211] Y. Wang, I. Chen, and J. Cho, “Trust-based task assignment in autonomous service-oriented ad hoc networks,” in *12th IEEE International Symposium on Autonomous Decentralized Systems*, Taichung, Taiwan, March 2015, pp. 71–77.

[212] Y. Wang, I. Chen, J. Cho, K. Chan, and A. Swami, “Trust-based service composition and binding with multi-objective optimization in service-oriented ad hoc networks,” *IEEE Transactions on Services Computing*, 2015, in press.

[213] Y. Wang, I. R. Chen, J. H. Cho, and J. J. P. Tsai, “Trust-based task assignment with multi-objective optimization in service-oriented ad hoc networks,” *IEEE Transactions on Network and Service Management*, vol. PP, no. 99, pp. 1–1, 2016.

[214] Y. Wang, A. Nakao, and A. V. Vasilakos, “Heterogeneity playing key role: Modeling and analyzing the dynamics of incentive mechanisms in autonomous networks,” *ACM Transactions on Autonomous and Adaptive Systems*, vol. 7, no. 3, Sept. 2012, article No. 31.

[215] U. Weerakoon and V. Allan, “Coalition formation for servicing dynamic motion tasks,” *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, Lecture Notes in Computer Science*, vol. 7874, pp. 386–393, 2013.

[216] R. E. Wendell and D. N. Lee, “Efficiency in multiple objective optimization problems,” *Mathematical Programming*, vol. 12, no. 1, pp. 406–414, 1977.

[217] A. K. Whitten, H. Choi, L. B. Johnson, and J. P. How, “Decentralized task allocation with coupled constraints in complex missions,” in *American Control Conference*, San Francisco, CA, June/July 2011, pp. 1642–1649.

[218] A. Wierzbicki, M. Makowski, and J. Wessels, Eds., *Model-Based Decision Support Methodology with Environmental Applications*. Dordrecht: Kluwer Academic Publishers, 2000.

[219] T. Xie and X. Qin, “An energy-delay tunable task allocation strategy for collaborative applications in networked embedded systems,” *IEEE Transactions on Computers*, vol. 57, no. 3, pp. 329–343, March 2008.

[220] K. Xu, K. C. Wang, R. Amin, J. Martin, and R. Izard, “A fast cloud-based network selection scheme using coalition formation games in vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5327–5339, Nov. 2015.

[221] L. Xu, Z. Zeng, and X. Ye, “Multi-objective optimization based virtual resources allocation strategy for cloud computing,” in *Proceedings of IEEE/ACIS 11th International Conference on Computer and Information Science*, Shanghai, China, May 2012, pp. 56–61.

[222] Z. Xu, Y. Li, and X. Feng, “Constrained multi-objective task assignment for UUVs using multiple ant colonies system,” in *Proceedings of ISECS International Colloquium on Computing, Communication, Control, and Management*, Guangzhou, China, Aug. 2008, pp. 462–466.

[223] B. Ye, Y. Wang, and L. Liu, “Crowd trust: A context-aware trust model for worker selection in crowdsourcing environments,” in *2015 IEEE International Conference on Web Services*, June 2015, pp. 121–128.

[224] P. Yin, S. Yu, P. Wang, and Y. Wang, “Task allocation for maximizing reliability of a distributed system using hybrid particle swarm optimization,” *Journal of Systems and Software*, vol. 80, no. 5, pp. 724–735, May 2007.

[225] J. Yu, M. Kirley, and R. Buyya, “Multi-objective planning for workflow execution on grids,” in *Proceedings of 8th IEEE/ACM International Conference on Grid Computing*, Washington DC, USA, Sept. 2007.

[226] J.-H. Yun and K. G. Shin, “Adaptive interference management of OFDMA femtocells for co-channel deployment,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 6, pp. 1225–1241, June 2011.

[227] M. Zamani, “Scalarization and stability in multi-objective optimization,” Ph.D. dissertation, University of Avignon and Vaucluse University of Tehran, 2016, general Mathematics.

[228] M. Zeleny, *Multiple Criteria Decision Making*, University of South Carolina, Columbia, 1973, ch. Compromise programming, pp. 262–301.

[229] J. Zhan, X. Fang, and P. Killion, “Trust optimization in task-oriented social networks,” in *IEEE Symposium*

on Computational Intelligence in Cyber Security, Paris, France, April 2011, pp. 137–143.

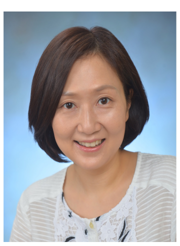
[230] H. Zhang, C. Jiang, N. C. Beaulieu, X. Chu, X. Wang, and T. Q. S. Quek, "Resource allocation for cognitive small cell networks: A cooperative bargaining game theoretic approach," *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3481–3493, June 2015.

[231] Q. Zhang, N. Mi, A. Riska, and E. Smirni, "Performance-guided load (un)balancing under autocorrelated flows," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 5, pp. 652–665, May 2008.

[232] W. Zhang and T. Gao, "A minmax method with adaptive weightings for uniformly spaced pareto optimum points," *Computers & Structures*, vol. 84, no. 28, pp. 1760–1769, 2006.

[233] Q. Zhu, H. Zeng, W. Zheng, M. di Natale, and A. Sangiovanni-Vincentelli, "Optimization of task allocation and priority assignment in hard real-time distributed systems," *ACM Transactions on Embedded Computing Systems*, vol. 11, no. 4, Dec. 2012, article 85.

[234] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proceedings of of Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, 2001.



Jin-Hee Cho received the BA from the Ewha Womans University, Seoul, Korea and the MS and PhD degrees in computer science from the Virginia Tech. She is currently a computer scientist at the U.S. Army Research Laboratory, Adelphi, Maryland. Her research interests include network security, trust and risk management, cognitive modeling, and network science. She received the best paper awards in IEEE TrustCom09 and BRIMS13. She is a recipient of the 2015 IEEE Communications Society William R. Bennett Prize in the field of Communications Networking, and a recipient of the 2013 Presidential Early Career Awards for Scientists and Engineers (PECASE). She is a member of ACM and a senior member of IEEE.

Yating Wang received her Ph.D degree from Virginia Tech in 2016. Her research interests include distributed networks, security, wireless networks, mobile computing, trust management, and reliability and performance analysis. Dr. Wang is currently a software engineer at Google.

Ing-Ray Chen received the BS degree from the National Taiwan University, and the MS and PhD degrees in computer science from the University of Houston. He is a professor in the Department of Computer Science at Virginia Tech. His research interests include mobile computing, wireless systems, security, trust management, and reliability and performance analysis. Dr. Chen currently serves as an editor for IEEE Communications Letters, IEEE Transactions on Network and Service Management, The Computer Journal, and Security and Network Communications. He is a recipient of the IEEE Communications Society William R. Bennett Prize in the field of Communications Networking. He is a member of IEEE.

Kevin S. Chan is a research scientist with the Computational and Information Sciences Directorate at the U.S. Army Research Laboratory (Adelphi, MD). Previously, he was an ORAU postdoctoral research fellow at ARL. His research interests are in network science, with past work in dynamic networks, trust and distributed decision making and quality of information through ARL's Network Science Collaborative Technology Alliance and Network and Information Sciences International Technology Alliance. Prior to ARL, he received a PhD in Electrical and Computer Engineering (ECE) and MSECE from Georgia Institute of Technology (Atlanta, GA). He also received a BS in ECE/EPP from Carnegie Mellon University (Pittsburgh, PA).

Ananthram Swami is with the US Army Research Laboratory as the Army's Senior Research Scientist (ST) for Network Science. Prior to joining ARL, he held positions with Unocal Corporation, USC, CS-3 and Malgudi Systems. He was a Statistical Consultant to the California Lottery, developed a Matlab-based toolbox for non-Gaussian signal processing. He has held visiting faculty positions at INP, Toulouse., and currently at Imperial College. He received the B.Tech. degree from IIT-Bombay; the M.S. degree from Rice University, and the Ph.D. degree from the University of Southern California (USC), all in Electrical Engineering. He is an ARL Fellow and a Fellow of the IEEE.