



# Energy-Delay Co-optimization of Resource Allocation for Robotic Services in Cloudlet Infrastructure

Mahbuba Afrin<sup>1(✉)</sup>, Jiong Jin<sup>1</sup>, and Ashfaque Rahman<sup>2</sup>

<sup>1</sup> School of Software and Electrical Engineering, Swinburne University of Technology,  
Melbourne, VIC 3122, Australia

{mafrin, jiongjin}@swin.edu.au

<sup>2</sup> Data61, CSIRO, Sandy Bay, TAS 7005, Australia  
ashfaque.rahman@data61.csiro.au

**Abstract.** Cloud based robotic services can be adopted for emergency management in smart factory. When multiple robots work collaboratively in such system, optimal resource allocation for executing the tasks of robotic services becomes a challenging problem due to the heterogeneity and energy consumption of resources. Since the tasks of multi-robotic services are inter-dependent, the inconvenience of data exchange between local robots and distant Cloud can significantly degrade the quality of service. Therefore, in this paper, we jointly address the energy consumption and service delay minimization problem while allocating resources in proximate Cloud (Cloudlet) based multi-robot systems for emergency management service in smart factory. A multi-objective evolutionary approach, NSGA-II algorithm is applied to solve this constrained multi-objective optimization problem. We augment the NSGA-II algorithm by defining a new chromosome structure, presorted initial population, mutation operator and selection of minimum distant solution from the non-dominated front to the origin while crossing over the chromosomes. The experimental results on the basis of synthetic data demonstrate that our approach performs significantly well compared to benchmark NSGA-II.

**Keywords:** Resource allocation · Multi-robot system · Cloudlet Optimization · Evolutionary algorithm

## 1 Introduction

In Cloud based multi-robot systems, Cloud offers virtualized resources, platform and software services so that both localized robots and remote resources can be utilized to process the tasks of robotic services. In a smart factory, such Cloud-robotic services can manage entire production and supply chain. However, safety assurance in smart factory during hazardous situation like fire occurrence is very crucial. In this case, both robot and Cloud resources should complement each other to process diversified tasks of emergency management service within a

stringent deadline. These tasks are usually inter-dependent, latency sensitive and compute intensive. In addition, the resources are heterogeneous in terms of processing capability and energy consumption [3]. In such scenario, an optimal allocation of computing resources to the tasks with the aim of minimizing both service delay and energy consumption is one of the key research challenges.

Taking cognizance of the aforementioned issues, we focus on energy-delay co-optimized resource allocation for processing the tasks of emergency management service such as fire driven emergency service in a smart factory across robot and Cloud infrastructure. Tasks of this service require real-time response and data transmission among robots and Cloud. Therefore, the concept of Cloudlet [4] infrastructure is introduced between robot-Cloud. Cloudlet is a prominent extension of Cloud closer to the data source that can provide multi-robot systems with virtualized resources to execute latency sensitive services.

To the best of our knowledge, this is the first work to design a Cloudlet based multi-robot framework for emergency management service in smart factory. Resource allocation in this context appears as a constrained multi-objective optimization problem when both energy consumption of resources and overall makespan for processing the tasks are targeted to minimize simultaneously. Since, multi-objective evolutionary algorithms help to generate pareto-optimal solutions of such multi-objective optimization problem, in this paper, we extend the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [2] to solve the resource allocation problem for its capability of finding diverse set of solutions. Here, we augment the NSGA-II by defining a new chromosome structure, pre-sorted initial population based on the task size and processing speed of the resources. Besides, while crossing over the chromosomes, rather than selecting arbitrarily, the chromosome having minimum distant solution from the pareto-front to the origin is selected to balance the values of both objectives in subsequent generations. The results of our simulation experiments significantly improves the benchmark NSGA-II in minimizing both objectives.

The rest of the paper is organized as follows. In Sects. 2 and 3, related work and Cloudlet-based framework for multi-robot system are discussed, respectively. The problem formulation and the solution mechanism are described in Sect. 4. In Sect. 5, the efficacy of our proposed approach is validated through simulation and Sect. 6 concludes the paper with future directions.

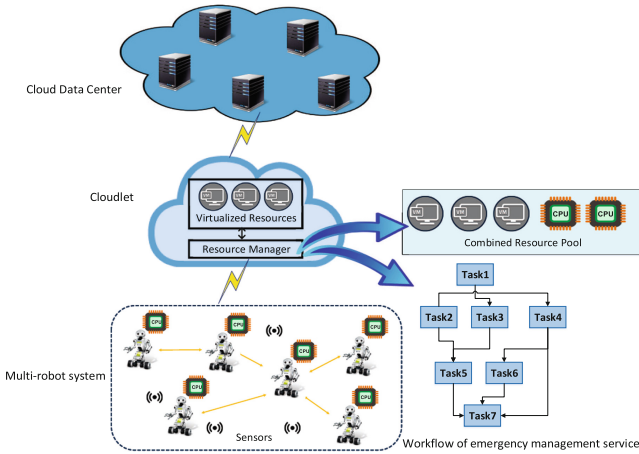
## 2 Related Work

In literature, the integration of Cloud services and multi-robot systems has already been investigated. The concept of Robot as a Service through a Service Oriented Architecture is defined in [1] that incorporates robot services in Cloud. A smart city-based Cloud robotic framework for optimal task offloading is designed in [7]. Motion and connectivity-aware offloading for Cloud robotic services using evolutionary algorithm is further introduced in [6]. However, in both [7] and [6], the offloading decision-making is done by a single robot.

Task allocation in resource-constrained multi-robot systems for search and rescue or other emergency and hazardous scenarios is discussed in [8]. Cost-efficient deployment of robots for a search and rescue use case during large-scale disaster management is explored in [5]. However, in the aforementioned works, no multi-robot task allocation policy targeting both local and Cloud resources is pursued. Nevertheless, considering the intrinsic constraints of Cloud and multi-robot assisted smart factory during the execution of emergency management services, neither the combined task allocation problem nor the solutions has been enlightened in the literature so far.

### 3 Framework for Cloudlet Based Multi-robot System

The Cloudlet infrastructure between robot and Cloud forms a three-tier computational framework and helps to execute emergency management service for smart factory with less communication delay. In this framework, the robots using Zigbee, Bluetooth and WiFi can communicate with each other. They also communicate with Cloudlet and Cloud through gossip protocols [3].



**Fig. 1.** Framework design for Cloudlet based multi-robot system

In a Cloudlet, *Resource Manager (RM)* creates a combined resource pool with both local and remote resources while executing the emergency management service as shown in Fig. 1. Here, we consider emergency fire management service in smart factory as an example case scenario. This service consists of multiple tasks such as fire origin and cause identification, human victim and hazardous material detection, evacuation planning, navigation as well as management of external help. These tasks are interdependent and orchestrated through a Directed Acyclic Graph (DAG) based workflow model. We assume that the corresponding

tasks of a service and the meta-data of the tasks (inter-data dependency delay, Quality of Service requirement etc.) are stored in the Cloudlet. Whenever an event of interest triggers the service initiation, the RM residing in the Cloudlet assigns the tasks to the combined resource pool.

## 4 Proposed Energy-Delay Optimized Resource Allocation

### 4.1 Multi-objective Optimization Problem Formulation

The optimal resource allocation for processing the service tasks is subject to multiple constraints driven by the characteristics of tasks (e.g., latency sensitivity, interdependency, resource requirements, etc.). It turns into a multi-objective optimization problem when simultaneous minimization of overall makespan for completing the tasks and energy consumption of the resources are taken into account. In the proposed Cloudlet-based framework, this multi-objective optimization problem can be expressed through Eqs. 1 and 2 for the corresponding set of tasks  $T$  and resources  $R$ . In Eq. 1,  $m_r^t$  denotes the time required to complete a task  $t \in T$  on a resource  $r \in R$  and it minimizes the overall makespan  $\forall t \in T$ . At the same time, Eq. 2 minimizes the overall energy consumption  $e_r^t$  while processing each task  $t \in T$  on particular resource  $r \in R$ . The solution of both equations points the assignment of task  $t \in T$  to resource  $r \in R$  through non-zero value of binary decision variable  $x_{tr}$ . The constraints of Eqs. 1 and 2 ensure that a task  $t \in T$  will not assign to multiple resources (Eq. 3), the total task completion time will not exceed the maximum allowable delay  $M_T$  to execute the service (Eq. 4) and the total energy consumption of resources should be within the energy threshold,  $E_T$  (Eq. 5). In addition, Eq. 6 signifies that the assignment of a dependent task  $t \in T$  to a resource  $r \in R$  and its tolerable inter-task data dependency delay  $\delta_t$  will not be affected by the assignment of all its predecessor tasks  $t' \in T'_t$  on computing resources  $r' \in R$ . In this case, maximum processing time  $p_{r'}^{t'}$  of the predecessor tasks and data exchange time  $\eta_{r'r}$ ,  $\forall t' \in T'_t$  is taken into account.

$$\min \sum_{t \in T, r \in R} x_{tr} \times m_r^t \quad (1)$$

and

$$\min \sum_{t \in T, r \in R} x_{tr} \times e_r^t \quad (2)$$

subject to

$$\sum_{t \in T, r \in R} x_{tr} = 1 \quad (3)$$

$$\sum_{t \in T, r \in R} m_r^t \leq M_T \quad (4)$$

$$\sum_{t \in T, r \in R} e_r^t \leq E_T \quad (5)$$

$$\max(p_{r'}^{t'} + \eta_{r'r}) \leq \delta_t; \forall t' \in T'_t \quad (6)$$

## 4.2 Energy-Delay Co-optimization Using Multi-objective Evolutionary Algorithm

To solve the resource allocation problem, we augment multi-objective evolutionary algorithm, NSGA-II. In NSGA-II algorithm, solution of a chromosome contains the values of different objectives obtained from fitness functions and the solutions of a population are classified into different sets according to the ascending level of their domination. A set of solutions  $S$  dominates another solution set  $S'$  if each solution  $x \in S$  is no worse than solution  $x' \in S'$ ,  $\forall x'$  in every objectives and each solution  $x \in S$  is strictly better than solution  $x' \in S'$ ,  $\forall x'$  in at least one objective. If a solution set is not dominated by any other set of solutions, the elements of that set is called the non-dominated solutions. Each set of solutions represents a particular front on the solution space and the chromosomes generating those solutions are treated as the builder of the front. The non-dominated set of solutions provides the optimal front (first front) on the solution space, is termed as the pareto-front. Here, we extend the basic concept of NSGA-II and further refine it to develop an energy-delay co-optimized resource allocation for emergency management service in Cloudlet infrastructure as follows:

**Population Initialization.** In the augmented NSGA-II, rather than creating randomized initial population, a presorted initial population relying on the heuristics is generated. Here, the set of resources  $R$  in the combined resource pool is divided into  $k$  categories based on the ascending processing speed  $\rho_r, \forall r \in R$ . Similarly, the set of tasks  $T$  is also classified in  $k$  types according to the incremental size  $\lambda_t, \forall t \in T$ . Thereafter, taking the problem range and constraints (Eqs. 3–6) into account, mathematical combination is used to conceptually assign the tasks of  $j$  type to the resources of  $j^{th}$  category for generating the chromosomes of initial population. The structure of chromosomes is aligned with the steps of initial population creation, where a Gene Index symbolizes a particular task and the Gene refers to a specific resource.

**Domination Count and Ranking.** The solution space of a population is determined by the outcome of its member chromosome's fitness value on makespan and energy objectives using Eqs. 1–2. The solutions of combined parent and child population  $U_i$  for any generation  $g_i; i \in \{1 \dots G\}$ , generated by its member chromosomes are ranked in different fronts within corresponding solution space based on the level of their domination according to NSGA-II [2].

**Selection of Population.** Since the size of combined population  $U_i$  for any generation  $g_i$  is  $2N$ , for iterative refinement, it is very important to select the best  $N$  number of chromosomes from  $U_i$  to form the parent population  $P_{i+1}$  for the next generation  $g_{i+1}$ . In this case, chromosome slots within  $P_{i+1}$  is filled with the builder chromosomes of comparatively better fronts on the solution space.

**Crowding Distance Calculation.** In selecting  $N$  number of chromosomes for the parent population  $P_{i+1}$  of next generation  $g_{i+1}$ , sometimes the available slot in  $P_{i+1}$  can be less enough to accommodate the entire builder chromosome set  $F_\tau$  of a particular front. In this case, crowding distance of the solutions are

calculated to identify the compatible builder chromosomes of that front to fill the available slot in  $P_{i+1}$ .

**Extension of Genetic Operator.** For a particular generation  $g_i$ , while generating the child population  $C_i$  from the parent population  $P_i$ , genetic operators such as fitness calculation, selection, crossover and mutation are applied. The fitness of the population is determined through the objective functions discussed in Eqs. 1–2. To imply mutation on the population, binomial distribution and to make crossover of a particular chromosome with the fittest chromosome of the population, simulated binary approaches are used. To select the fittest chromosome, for each chromosome  $c$ , the distance  $c.d$  of makespan and energy consumption objective value ( $c.M, c.E$ ) from the theoretical lowest value of makespan and energy,  $(0, 0)$  is calculated. The chromosome having minimum distance value from the origin is selected as the fittest chromosome of the population for making crossover. However, the selection of final solution and characteristics of the corresponding chromosome solely depends on the intention of service providers.

## 5 Simulation Results and Discussion

### 5.1 Simulation Environment

To conduct our experiments in Matlab, we use synthetic data, driven from real-world references and selects system parameters carefully for fair evaluation [6]. The parametric values for the simulation environment are summarized in Table 1. The value of simulation parameters within a specific range is set by a pseudo random number generator. For simplicity of the simulation, we consider three types of computing resources based on their heterogeneous processing speed. In addition, we set the ratio of local and virtual resources in the combined resource pool to  $\frac{1}{3}$  and the ratio of dependent and independent tasks is set to  $\frac{2}{5}$ .

### 5.2 Simulation Scenarios and Result Analysis

The performance of our proposed approach is compared with a benchmark strategy which follows NSGA-II algorithm with randomized initial population and arbitrary chromosome selection during crossover.

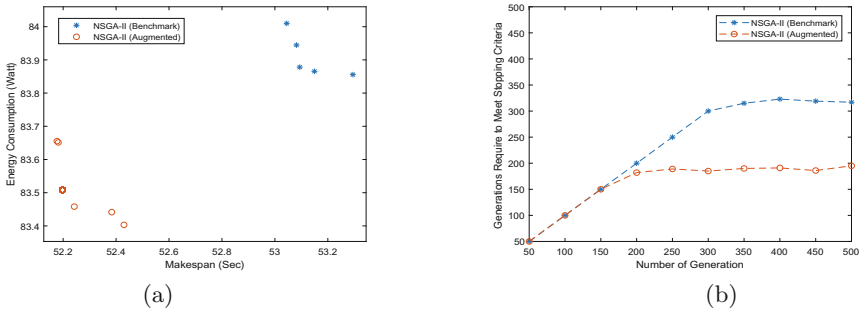
**Comparison of Pareto-Optimal Solutions and Generations to Meet Stopping Criteria.** After 200 generations, the pareto-optimal solutions of our augmented NSGA-II and benchmark NSGA-II on fixed number of heterogeneous tasks (50) and resources (30) are depicted in Fig. 2(a). In this scenario, each pareto-optimal solutions of energy-delay co-optimized approach provides better outcome for both objectives, compared to the benchmark strategy. The initial population of proposed approach that is determined based on the task’s size and processing speed of resources, inherently minimizes the overall makespan and energy consumption. In addition, selection of the chromosome having minimum distant solution from the origin for crossover further improves its performance.

**Table 1.** Simulation parameters

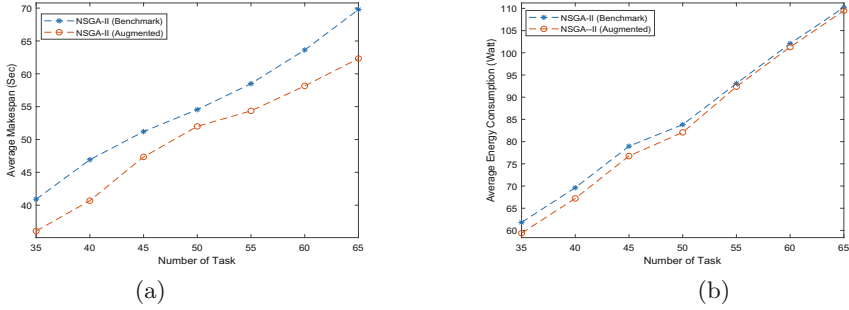
Parameter	Value
Population size	50
No of generations	50–500
Mutation rate	0.5
Crossover rate	0.5
Number of tasks in a service	35–65
Number of computing resources	15–45
Processing speed of virtual resources	10000–30000 MIPS
Processing speed of local resources	8000–15000 MIPS
Per unit energy consumption of virtual resources	50–150 W
Per unit energy consumption of local resources	40–90 W
Tasks data size	5000–10000 MI
Allowable completion time of all tasks	5000 ms
Maximum allowable energy consumption of service	2500 W
Data dependency threshold	1500 ms
Communication bandwidth	128–512 Kbps

Therefore, after a fixed number of generations, it provides significantly better solution than the randomized benchmark. The simulation results on Fig. 2(b) represents the efficacy of proposed approach in achieving no further optimization state within less generations compared to the benchmark strategy. For using pre-sorted population, compared to the benchmark, our approach implicitly advances a certain number of generations in meeting the stopping criteria.

**Impacts of Varying Number of Tasks.** The average makespan and energy consumption for varying number of tasks while the number of generations (200)



**Fig. 2.** (a) Pareto-optimal solutions and (b) Number of generations required to meet stopping criteria



**Fig. 3.** (a) Average makespan and (b) average energy consumption of resources for varying number of tasks

and resources (30) are fixed, are illustrated in Figs. 3(a) and (b) correspondingly. Since the resource number is fixed, with increasing of the tasks objective values increases for both approaches. However, compared to benchmark, our proposed approach provides improved solutions in this scenario due to presorted population. In the presorted population, the computing resources are categorized based on their processing speed for assigning the tasks to a particular type of computing resources according to their size. As a consequence, it generates efficient initial population and refines it consistently after each generation.

## 6 Conclusion and Future Directions

Evaluation results acquired through simulating different scenarios show that our augmented NSGA-II algorithm improves the benchmark NSGA-II algorithm by minimizing both makespan and energy consumption. In future, our proposed approach will be compared with other existing multi-objective problem solving approaches like PAES, SPEA. Furthermore, as our evaluation is based on simulation, sources of uncertainty in the system and their effects on simulation parameters, sensitivity of independent variables will be analysed and implemented in real-world testbed. The impact of complexities on different tasks will also be investigated. Utilizing edge resources for task allocation will be a good extension.

**Acknowledgement.** The authors would like to thank Data61, CSIRO, Australia for supporting the work.

## References

1. Chen, Y., Du, Z., García-Acosta, M.: Robot as a service in cloud computing. In: 5th IEEE International Symposium on Service Oriented System Engineering, pp. 151–158. IEEE Computer Society, Washington (2010)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)



3. Hu, G., Tay, W.P., Wen, Y.: Cloud robotics: architecture, challenges and applications. *IEEE Netw.* **26**(3), 21–28 (2012)
4. Mahmud, M.R., Afrin, M., Razzaque, M.A., Hassan, M.M., Alelaiwi, A., Alrubaiyan, M.: Maximizing quality of experience through context-aware mobile application scheduling in cloudlet infrastructure. *Softw.: Pract. Exp.* **46**(11) (2016)
5. Mouradian, C., Yangui, S., Glitho, R.H.: Robots as-a-service in cloud computing: search and rescue in large-scale disasters case study. *CoRR* (2017)
6. Rahman, A., Jin, J., Cricenti, A., Rahman, A., Panda, M.: Motion and connectivity aware offloading in cloud robotics via genetic algorithm. In: *IEEE Global Communications Conference*, pp. 1–6 (2017)
7. Rahman, A., Jin, J., Cricenti, A., Rahman, A., Yuan, D.: A cloud robotics framework of optimal task offloading for smart city applications. In: *IEEE Global Communications Conference*, pp. 1–7 (2016)
8. Wu, D., Zeng, G., Meng, L., Zhou, W., Li, L.: Gini coefficient-based task allocation for multi-robot systems with limited energy resources. *IEEE/CAA J. Autom. Sin.* **5**(1), 155–168 (2018)