Modelling Energy-Aware Task Allocation in Mobile Workflows

Bo Gao(✉) and Ligang He(✉)

Department of Computer Science, University of Warwick,
Coventry CV4 7AL, UK
{bogao,liganghe}@dcs.warwick.ac.uk

Abstract. Mobile devices are becoming the platform of choice for both business and personal computing needs. For a group of users to efficiently collaborate over the execution of a set workflow using their mobile devices, the question then arises as to which device should run which task of the workflow and when? In order to answer this question, we study two common energy requirements: in the *minimum group energy cost problem (MGECP)* we build the model as a quadratic 0–1 program and solve the optimisation problem with the objective to minimise the total energy cost of the devices as a group. In the *minimum max-utilisation problem (MMUP)* we aim to improve the fairness of the energy cost within the group of devices and present two adjustment algorithms to achieve this goal. We demonstrate the use of a Mixed Integer Quadratic Programming (MIQP) solver in both problem's solutions. Simulation result shows that both problems are solved to good standards. Data generated by different workload pattern also give us a good indication of the type of workflow that benefit the most from MMUP. The model used in this work can also be adapted for other energy critical scenarios.

Keywords: Mobile computing · Energy-aware · Collaboration · Workflow

1 Introduction

Recent years have seen significant growth in the size of the mobile computing market, and yet the rarest commodity in the world of mobile computing remains to be its battery power. Development in battery technology is slow compared to other components of a mobile smart device. Hence, despite the moderately improved battery capacity on modern smart devices, with increasingly more complex functionality required from the user, developments of mobile applications remain largely energy-constrained [17].

In less than a decade, mobile devices have enriched their functionalities from being a simple dialling device to a hub of rich media applications. It is predicted that by 2015, mobile application development projects will outnumber desktop projects by a ratio of 4:1 [10]. The unique portability of a mobile device coupled with its ever growing hardware capability brings business and ad hoc workflows

I. Stojmenovic et al. (Eds.): MOBIQUITOUS 2013, LNICST 131, pp. 89–101, 2014.
DOI: 10.1007/978-3-319-11569-6_8

that are traditionally supported by fixed location resources to be implemented over wireless mobile platforms.

Researches show that in a mobile environment, communication tasks are especially energy-demanding compared to local computation tasks [16]. Hence, this type of applications, namely *mobile workflows*, which has a particular emphasis on collaboration between users, is likely to be more energy-demanding than others and requires to be managed in an energy-efficient manner. Furthermore, unlike its desktop counterparts, mobile computing devices are often exposed to the open environment. Changing conditions in data connection, sudden drain of battery caused by user actions can bring disruption to a device's availability.

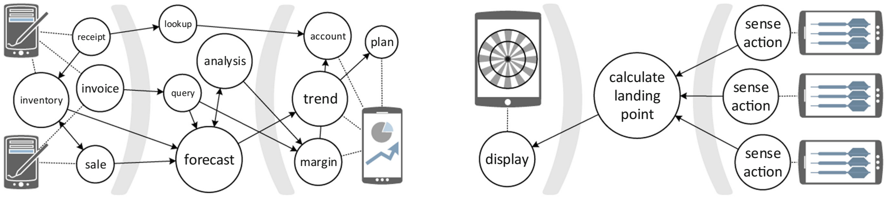
Our research investigate ways to model and analyse the energy efficiency of such workflows running atop a group of mobile devices. Our goal is to provide an energy efficient execution platform for mobile workflows, while utilising fair share of each mobile device's energy. Our objective is two-fold: First, in the MGECP, we aim to minimise the workflow's total energy consumption. Second, in the MMUP, our objective is to minimize the maximum device utilisation in the group while keeping the overall energy cost close to the minimum.

In this paper, we first give use cases from possible application areas of a mobile workflow and discuss related work. We then construct the system energy model in Sect. 3. The allocation problem is modelled as a quadratic 0–1 program, and its two objectives (MGECP and MMUP) are studied in Sects. 4 and 5. We conclude the paper in Sect. 6 with a simulation study the result of which verifies our formulation and compares the results of our algorithms when applied to different types of workflows.

2 Applications and Related Work

Mobile workflow can be found when a group of mobile users are to share or communicate with each other in order to accomplish a certain task. Such scenario commonly exists in a business environment. With growing adaptation of mobile devices within their business models [10], modern *enterprise applications* often include or are entirely based on mobile devices. For instance, in a supply chain business, as illustrated in Fig. 1a, the commencement of a workflow is triggered by a member of staff registering receipt/sales of goods on their mobile devices. The system database is then updated via a query module[1]. A forecast module is then evoked to produce a forecast based on the update, which is then projected onto the manager's smartphone as a live trend graph or a production plan. In order to lower the overall energy cost of the workflow, our objective in Sect. 4, the forecast module which requires complex computation for data mining purposes is more suitable to be allocated to a device that has a fast processor and low energy draw while running computation tasks. Additionally, it is preferable that modules that communicate frequently, e.g. invoice and query, are allocated to the same device to reduce communication cost.

[1] E.g. we can assume that the support system is similar to that of an Excel application with embedded VBA macro modules. Data is stored in the local spreadsheet.



(a) Workflows in enterprise application (b) Workflow in consumer application

Fig. 1. Examples of mobile workflow use cases. In the centre of both figures, the tasks between the pair of “) (” are not restricted to be executed on any specific device. Allocation of these tasks can affect the energy-efficiency of the workflow.

Another use case illustrated in Fig. 1b includes the use of three smartphones and a tablet, and expands on the idea of a popular *consumer application* [12] which lets its users to play darts with their mobile devices. During the game, a tablet is used to display the dartboard, participating players use their smartphones as darts. The workflow starts when a player throws a dart (by a throwing gesture from the phone towards the tablet). Sensor readings (accelerometer and gyroscope) are then taken from the phone and fed into a calculation module to work out where the dart should land on the board. Result from the calculation is then passed on to the display module on the tablet.

Like all multi-player competitions, the game can only function until its weakest player withdraws, which in this case, is the device that runs out of battery first. Although the calculation module does not require much energy at each run, repeated execution is required. As the game goes on, the battery of the device to which the calculation task is allocated drains faster than the others’. A fair task allocation, which we study in Sect. 5, is needed in such scenarios to balance the contribution made by participating members of the mobile workflow.

A workflow engine is often required to oversee the execution of mobile workflows. In [15] a detailed mobile workflow engine is implemented and tested on Nokia devices. A decentralised workflow coordination architecture designed for mobile devices is presented in [1] for use in biological studies and the supply-chain industry. Authors of [14] propose a rapid application development framework based on a dynamic workflow engine for creating mobile web services.

Several researches has been carried out in workflow management issues in Mobile *Social Content Sharing applications* [6,11,18]. A mobile P2P social content sharing framework was proposed in [6]. In [18], a Java API based mobile workflow system was proposed. A content distribution protocol was proposed in [13] for vehicular ad hoc networks (VANET). Clusters of mobile devices has been proposed in [22] to support the execution of parallel applications.

The common approach towards an allocation problems often model the problem as a linear program [7,19,21]. A linear program is suitable for modelling situations where communication time is not considered or when there are only two devices involved in the process. However, in the cases of mobile workflows, communication tasks are an essential part of the workload and occurs significant amount of energy cost [20]. Thus we construct our model as a quadratic program in order to accurately capture the communication costs.

Several recent researches has developed methods to measure the energy cost of mobile applications [8,16,20]. The difference in the current draw between sender and receiver in a wireless network can be read at [9]. Reference [3] includes a detailed characteristics of a WiFi network's energy pattern. Our energy model draws ideas from these researches.

3 System Model

3.1 Mobile Platform Model

We consider a mobile platform *MP* consisting of *m mobile devices*, $M_1, \cdots, M_m$, and denote a device profile as $M_i\left(s_i, e_i^{cmp}, e_i^{snd}, e_i^{rcv}\right), i \in \{1, 2, \ldots, m\}$ with parameters defined as follows:

s_i	Peak processing speed of M_i, measured in the number of clock cycles available in a millisecond;
e_i^{cmp}	Current draw from the battery when the device is executing computation tasks at peak speed;
$e_i^{snd/rcv}$	Current draw from the battery when the device is sending/ receiving data to/from the data network.

These devices are interconnected via a network, and we use b_{ij} to denote the bandwidth between devices M_i and M_j, $i, j \in \{1, 2, \ldots, m\}$. Thus, we have an m-matrix $B = (b_{ij})_{m\times m}$ which holds all of the bandwidth information of the underlying network of the *MP*. When two adjacent tasks are assigned to the same device, we assume that they share the same memory address space on the device. Therefore, we assign positive infinite values to the principal diagonal elements of B, that is $b_{ii} = +\infty, i \in \{1, 2, \ldots, m\}$.

3.2 Workflow Model

The workflow hosted on *MP* is represented by a directed acyclic graph $W = (T, R)$ whose vertex set $T = \{t_1, \ldots t_n\}$ denotes the set of *tasks* of the workflow. We assume that all tasks are defined via a service-oriented architecture and that all services are available from each device. An n-matrix $D = (d_{a,b})_{n\times n}$ denotes the weighted adjacency matrix of W, where $d_{a,b}$ is the size of the data package that is to be sent from t_a to t_b for $(t_a, t_b) \in R$. The acyclic property of W implies that D has all principle diagonal elements zero.

Each task has profile $t_a\left(d_{(.a)}, d_{(a.)}, c_a\right)$, $a \in \{1, \ldots n\}$ where $d_{(.a)}$ and $d_{(a.)}$ are the a-th column and the a-th row of D which represent the incoming and outgoing data respectively. c_a denotes the size/workload of the task.

3.3 Mobile Energy Model

Given an allocation scheme $\psi : T \to M$, we first derive the energy cost of computing $t_a, a \in \{1, \dots n\}$ to be

$$\mathcal{E}^{cmp}_{a\psi(a)} = e^{cmp}_{\psi(a)} \times \frac{c_a}{s_{\psi(a)}} \tag{1}$$

where $\psi(a)$ is the device to which t_a is assigned. Secondly, we have the energy cost of transferring d_{ab}, $(t_a, t_b) \in R$ as

$$\mathcal{E}^{tran}_{ab\psi(a)\psi(b)} = \underbrace{e^{snd}_{\psi(a)} \times \frac{d_{ab}}{b_{\psi(a)\psi(b)}}}_{\text{sender's cost}} + \underbrace{e^{rcv}_{\psi(b)} \times \frac{d_{ab}}{b_{\psi(a)\psi(b)}}}_{\text{receiver's cost}} \tag{2}$$

4 Minimum Group Energy Cost Problem (MGECP)

In this section, we first show that the Minimum Group Energy Cost Problem can be modelled as a generalised Quadratic Assignment Problem (QAP) [5] and then we convexify the objective function in order to solve it using a MIQP solver.

To represent an allocation scheme ψ, we first construct an $n \times m$ binary matrix $X = (x_{ai})$, such that

$$x_{ai} = \begin{cases} 1 & \text{if } \psi(a) = i, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We call matrix X an *assignment matrix* and a valid assignment must satisfy the following constraints

$$\sum_{i=1}^{m} x_{ai} = 1, \quad a = 1, 2, \dots, n, \tag{4}$$

$$x_{ai} \in \{0, 1\}, \quad a = 1, 2, \dots, n, \;\; i = 1, 2, \dots, m. \tag{5}$$

(4) ensures that every task must be assigned to one and only one device. (5) states that all tasks are indivisible.

4.1 Quadratic Program Formulation

With (1) (2) and (3), we can derive the total energy cost function as

$$\sum_{b=1}^{n}\sum_{j=1}^{m}\sum_{a=1}^{n}\sum_{i=1}^{m} \left(e^{snd}_i + e^{rcv}_j\right) \frac{d_{ab}}{b_{ij}} x_{ai} x_{bj} + \sum_{a=1}^{n}\sum_{i=1}^{m} e^{cmp}_i \frac{c_a}{s_i} x_{ai} \tag{6}$$

The quadratic terms in (6) gives the total energy cost for data transmission, whereas the linear term gives the total energy cost for executing computing tasks. We introduce $(nm)^2$ coefficients q_{aibj}

$$q_{aibj} := \begin{cases} e_i^{cmp} \dfrac{c_a}{s_i} + \left(e_i^{snd} + e_j^{rcv}\right) \dfrac{d_{ab}}{b_{ij}} & \text{if } (a, i) = (b, j), \\ e_i^{snd} \dfrac{d_{ab}}{b_{ij}} & \text{a } < \text{ b} \\ e_i^{rcv} \dfrac{d_{ba}}{b_{ij}} & \text{a } > \text{ b} \end{cases} \tag{7}$$

and with (7) we can transform (6) to

$$\sum_{b=1}^{n} \sum_{j=1}^{m} \sum_{a=1}^{n} \sum_{i=1}^{m} q_{aibj} x_{ai} x_{bj} \tag{8}$$

Theorem 1. *Let coefficients q_{aibj} be the entries of an $mn \times mn$ matrix Q, such that q_{aibj} is on row $(i-1)\,n + a$ and column $(j-1)\,n + b$, and $x = vec(X) = (x_{11}, x_{12}, \ldots, x_{1n}, x_{21}, \ldots, x_{mn})^T$ be the vector formed from the columns of X. Equivalent formulations for the minimum workflow energy cost problem's objective function are given by (8) and*

$$vec\,(X)^T\, Q\; vec\,(X) \tag{9}$$

Proof. From the construction of $vec\,(X)$, we observe that its u-th element $vec\,(X)_u = x_{ai} \Leftrightarrow u = (i-1)\,n + a$. Furthermore, given $u = (i-1)\,n + a$ and $v = (j-1)\,n + b$, $u, v \in \{1, 2, \ldots, mn\}$, we also get $Q_{uv} = q_{aibj}$. Hence,

$$\begin{aligned} (8) &= \sum_{v=1}^{mn} \sum_{u=1}^{mn} vec\,(X)_u^T\, Q_{uv}\; vec\,(X)_v \\ &= \sum_{b=1}^{n} \sum_{j=1}^{m} \sum_{a=1}^{n} \sum_{i=1}^{m} x_{ai} q_{aibj} x_{bj} = (9) \end{aligned}$$

□

4.2 Convexification

In order to exploit the power of modern MIQP solvers, we first need to preprocess the problem and convexify the objective function [4]. There are a number of ways of convexification. Our process is similar to that use in [2].

Theorem 2. *Let $Q^* := 1/2\left(Q + Q^T\right) + \alpha I$, where I is the $mn \times mn$ identity matrix, then Q^* is positive definite if scalar $\alpha = 1 + \parallel Q \parallel_\infty$*

Proof. Due to the length of this paper, interested reader are referred to the appendix of [2] (on a negative definite matrix) for a similar proof. □

Addition of a constant on the main diagonal of Q only add a constant to (9) which does not change its optimal solution. Hence we can rewrite our objective function as

$$\text{min: } vec\,(X)^T\, Q^*\; vec\,(X) \tag{10}$$

This together with (4) and (5) completes the formulation of the optimisation problem of MGECP. The positive definite property of Q^* ensures that (10) is strictly convex and a global minimum can be found by an MIQP solver.

5 Minimum Max-Utilisation Problem (MMUP)

While MGECP ensures that a workflow consumes minimum amount of energy from the mobile devices as a group, it does not consider the stress it has on individual devices. This causes unfair energy cost distribution within the MP, and creates *over-utilised* devices. Having such workflow executed repeatedly over time without adjustment to its task allocation scheme could lead to early retirement of the over-utilised devices. In a business environment, it is common to have authorisation constrained tasks taking critical roles within workflows. In such cases, the MP's inclusion of these authorised devices is critical to the fulfilment of the workflow's functionality. This requires the workflow engine to shift its priority from reducing the total energy cost of the device group to ensuring the availability of key devices.

Hence in this section of the paper, we investigate ways to adjust the task allocation provided by the MGECP so that the availability period of a workflow can be lengthened. We refer to this class of problem as the Minimum Max-Utilisation Problem (MMUP). We first introduce the measure of utilisation:

Definition 1. *Given an allocation scheme ψ, the* utilisation *of M_i, denoted $\mathcal{U}_i^{\psi}$, equals $\mathcal{E}_i^{\psi}/\mathcal{E}_i^{R}$, for $1 \leq i \leq m$, where $\mathcal{E}_i^{\psi}$ is the energy cost of M_i under ψ and $\mathcal{E}_i^{R}$ is the size of the residual energy in M_i.*

The reciprocal of a device's utilisation, $(1/\mathcal{U}_i)$, is the number of times the workflow can run with M_i before it runs out of battery. The availability of a workflow is hence constrained by the member with the highest value of utilisation. As illustrated in Fig. 2, we introduce a *guide utilisation* value $\mathcal{U}^G$ to classify devices into two groups: *Over-Utilised (OU)* and *Under-Utilised (UU)*. The objective then is to shift workload from devices in the OU group to those in UU.

We present two adjustment methods, both utilising the quadratic program formed in MGECP and use the result it produces to apply tight constrains to both methods' variables so that the group's overall energy cost remains minimised to a good degree. Upon need, or periodically, the workflow engine executes the adjustment algorithm in order to map the workflow to an updated task allocation scheme so that no device is over stressed unnecessarily and thus improve the availability of the workflow.

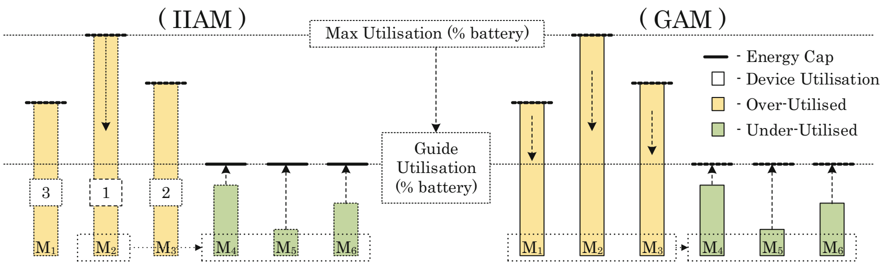


Fig. 2. MMUP adjustment algorithms

For both adjustment methods, in order to constraint each device's energy cost, we introduce a device specific cost matrix Q^i as an addition to the quadratic program formulated in MGECP.

Theorem 3. *Let*

$$Q^i_{uv} = \begin{cases} Q_{uv} & \textit{if } n \times (i-1) < u \leq n \times i, \\ 0 & \textit{otherwise.} \end{cases} \tag{11}$$

Then given an allocation scheme ψ and its allocation matrix X^ψ, we have the energy cost of M_i to be $\mathcal{E}_i^\psi = vec\left(X^\psi\right)^T Q^i \, vec\left(X^\psi\right)$

Proof. Proof is similarly to that of Theorem 1 and can be worked out easily. □

5.1 Iterative Individual Adjustment Method (IIAM)

In this method, we aim to reduce the energy cost of devices in OU individually (as illustrated in Fig. 2). With (11), we formulate a quadratically constrained quadratic program (QCQP) with an objective function that minimise the energy cost of the device with the highest utilisation value. As constraints in the QCQP, we cap all other OU devices' energy cost to their current value and all UU devices to the guide utilisation value. (For brevity we use the average utilisation of the current allocation scheme as our guide value. This can be replaced with tailored values to suit the requirement of certain workflows).

If the solver returns a new allocation, we then update the OU and UU group and again select the highest utilised device to the objective function. If this device is same to the one we picked at the earlier iteration, this means that we have reached the optimum solution under the constraints and exit. Otherwise, we repeat the process with the updated group classification until no new device can be picked from the OU group and provide a new allocation.

The advantage of this method is that it pin-points the highest utilised device of the MP, and support its workload offload with the entirety of UU devices. The disadvantage of this method is that all other members of the OU group is capped at their current utilisation value, this restraints the workload offload on the objective device when communication tasks exist between these devices.

5.2 Group Adjustment Method (GAM)

Similar to IIAM, our second adjustment method also caps the contribution from the UU devices at the guide value which ensures that the relocation process does not produce a new OU device (as illustrated in Fig. 2). It also apply cap to all OU devices to their current value. Unlike IIAM, the objective function in GAM include all devices in the OU group and also do not iterate.

The advantage of GAM over IIAM is that the workload offload is done between two groups of devices and thus encourages workload offload between members of each group. This in turn increase the possibility of producing a new allocation. The main disadvantage is that it does not prioritise on reducing the maximum utilisation which the workflow's availability is limited to.

6 Simulation

6.1 Environmental Settings

While it is intractable to cover all possible use cases of mobile workflows, we aim to base our simulation closely to the characteristics of an average mobile application and a modern smart device. We construct our simulation with the multiples of two essential building blocks: a *typical device* and a *unit workload*.

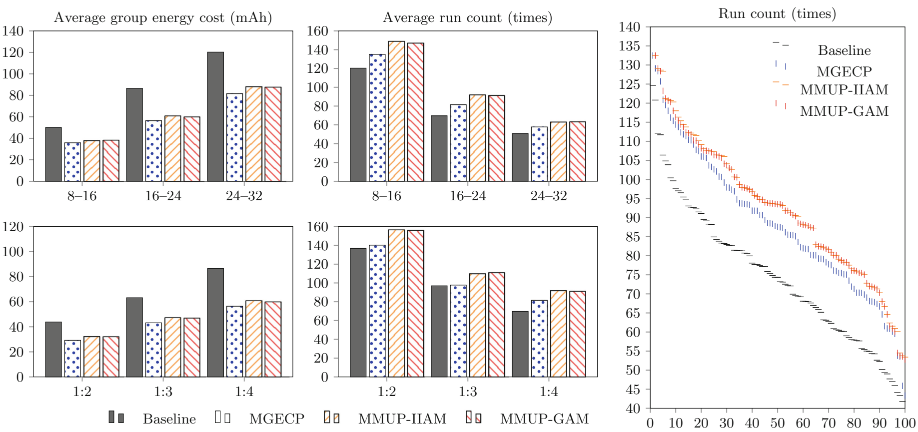
Definition 2. *A* typical mobile device *has a battery capacity of 2000 mAh, draws a current of 250–400 mA during data transmission and 100–200 mA when executing local computation tasks.*

In order to accurately emulate the correlation between modern mobile applications and the behaviour and capability of a state-of-the-art smart device, we consult the data presented in recent researches [8, 16, 20].

Definition 3. *A task has a* unit workload *if its execution takes 1 s to complete on a typical device.*

In our simulation, we specify each task's workload size using multiples of a unit workload. For instance, in the first plot of Fig. 3a, the tests are in 3 groups and the workflow generated in each test group has a task size that ranges in from 8 to 16, 16 to 24 and 24–32 units of a unit workload.

Apart from task size, many other factors (e.g. network bandwidth, etc.) also affects the energy cost of a workflow. Due to the length of this paper, we select to present the effect of different device to task ratios in our simulation settings to further verify our model and adjustment methods, as shown in the latter two plots of Fig. 3a. We also present the effect of different workload size distribution



(a) Lower group energy cost and higher run count.

(b) Sorted run counts

Fig. 3. Reduction in group energy cost and increase in workflow run count

Table 1. Comparison of adjustment methods

Workloads‡	Tests	MMUP-IIAM (MGECP) No.	Max.†	Avg.†	Dev.†	MMUP-GAM (MGECP) No.	Max.†	Avg.†	Dev.†
Exp. 6	100	35	0.311(0.332)	0.210(0.192)	0.069(0.103)	29	0.295(0.320)	0.204(0.191)	0.062(0.098)
Uni. 4-8	100	41	0.329(0.360)	0.229(0.214)	0.068(0.105)	38	0.334(0.367)	0.230(0.214)	0.068(0.106)
Exp. 8	100	33	0.399(0.436)	0.266(0.242)	0.092(0.141)	30	0.403(0.437)	0.266(0.244)	0.093(0.141)
Uni. 4-12	100	45	0.440(0.486)	0.308(0.287)	0.087(0.140)	42	0.440(0.488)	0.306(0.289)	0.087(0.140)
Exp. 12	100	29	0.651(0.728)	0.425(0.391)	0.151(0.238)	31	0.628(0.706)	0.420(0.391)	0.137(0.225)
Uni. 4-20	100	44	0.640(0.716)	0.464(0.427)	0.124(0.221)	42	0.646(0.727)	0.452(0.424)	0.131(0.226)
Exp. 20	100	26	0.987(1.050)	0.640(0.598)	0.243(0.329)	26	1.014(1.082)	0.651(0.601)	0.250(0.349)
Uni. 4-36	100	49	1.113(1.244)	0.773(0.714)	0.234(0.378)	50	1.097(1.202)	0.754(0.701)	0.226(0.365)

† - All utility values are percentages (%) of residual battery (mAh). ‡ - Distribution and task size.

pattern in Table 1. For each simulation setting 100 instances are randomly generated and worked on. Averages are taken for comparison. We use AMPL and CPLEX 12.5's MIQP solver to solve the formulated problems.

6.2 Results and Analysis

Minimum Total Energy Cost. The first group of our simulations aims to verify the formulation of MGECP. As a comparison, we use a baseline algorithm which attempts to reduce the total energy cost by distributing the number of tasks evenly across the MP. This algorithm provides a good baseline value because although it does not seek the benefit of using an energy efficient device, its chance of being able to take that advantage is consistent.

As shown in Fig. 3a, the total energy cost of a baseline allocation is reduced by 30–35% with MGECP applied. Both adjustment algorithms are applied to allocation produced by MGECP. As shown in Fig. 3a, the adjustments does not significantly increase the total energy cost of the workflow, but boost the workflow's run count (the number of time the workflow can run before the first retirement from the MP). As discussed with the example illustrated in Fig. 1b, the fair distribution of workload amongst the MP is critical. One series of simulation (16–24 in Fig. 3a) is magnified and plotted in Fig. 3b to illustrate the effect our algorithms have in extending the run count of workflows.

Utilisation Adjustment. This group of simulations focuses on the adjustment algorithms and their effect on workflows with different task workload range and distribution. Results (cf. Table 1) show that first of all, not all MGECP allocations can be adjusted because of the tight constraints we apply in both IIAM and GAM. 26–50% of the 100 test instances generated in each setting can be adjusted in order to gain a lower maximum utility.

It is worth noting that workflows with uniformly distributed task sizes have better chance to be adjusted than those with exponential distribution pattern. Adjustments can be made when tasks can be offloaded or exchanged between

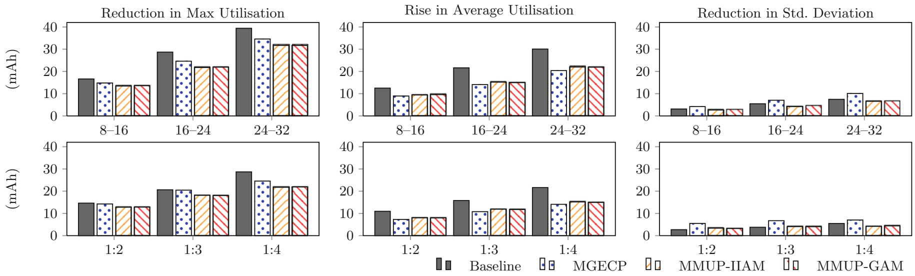


Fig. 4. Effect of adjustments within the MP.

devices without causing sizeable disturbance in each devices' energy cost. Contrary to that of exponential distribution, when workload's distribution is uniform within a set range, it is likely for a task to find another task that has similar workload size, thus a "minor" exchange of tasks is more likely to exist.

Device Energy Cost. This group of simulation focuses on the energy cost of individual devices. Figure 4 shows that in order to reduce the maximum utilisation, tasks has to be offloaded or exchanged to a device where it will cost more energy to execute which increase the average utilisation of the group. Increase in the group's standard deviation caused by MGECP shows that in order to minimise the group's energy cost, devices with better energy-efficiency are required to take on more workload than the others. On the other hand, the reductions of this value from MGECP to MMUP show that the workflow's energy cost is distributed more evenly within the MP after adjustments.

7 Conclusion

In this paper, we introduced a model that captures both computation and communication costs of a workflow with a 0–1 quadratic program. We demonstrated the use of MIQP solvers which produces exact solutions for the MGECP. We also investigated ways to adjust the allocation to lengthen the workflow's availability with minimal impact on its overall energy cost. Our simulation produces good results for both problems and gives an insight into workflows of different characteristics. Our model is also applicable to other energy critical scenarios, its extension can be tailored for workflows of specific use cases.

Acknowledgement. This work is sponsored by the Research Project Grant of the Leverhulme Trust (Grant No. RPG-101).

References

1. Balasooriya, J., Joshi, J., Prasad, S.K., Navathe, S.: Distributed coordination of workflows over web services and their handheld-based execution. In: Rao, S., Chatterjee, M., Jayanti, P., Murthy, C.S.R., Saha, S.K. (eds.) ICDCN 2008. LNCS, vol. 4904, pp. 39–53. Springer, Heidelberg (2008)
2. Bazaraa, M.S., Sherali, H.D.: On the use of exact and heuristic cutting plane methods for the quadratic assignment problem. J. Oper. Res. Soc. **33**(11), 991–1003 (1982)
3. Bejerano, Y., Han, S.J., Li, L.E.: Fairness and load balancing in wireless LANs using association control. In: MobiCom'04 Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, p. 315 (2004)
4. Billionnet, A., Elloumi, S.: Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. Math. Program. **109**(1), 55–68 (2006)
5. Burkard, R.E., Pitsoulis, L.S., Linearization, J., Polytopes, Q.A.P.: The quadratic assignment problem. In: Pardalos, P.P., Resende, M.G.C. (eds.) Handbook of Combinatorial Optimization. Kluwer Academic Publishers, Dordrecht (1998)
6. Chang, C., Srirama, S.N., Ling, S.: An adaptive mediation framework for mobile P2P social content sharing. In: Liu, C., Ludwig, H., Toumani, F., Yu, Q. (eds.) ICSOC 2012. LNCS, vol. 7636, pp. 374–388. Springer, Heidelberg (2012)
7. Cuervo, E., Balasubramanian, A., Cho, D.k., Wolman, A., Saroiu, S., Chandra, R., Bahl, P.: MAUI: making smartphones last longer with code offload. In: MobiSys'10 the 8th International Conference on Mobile Systems, Applications, and Services (2010)
8. Dong, M., Zhong, L.: Self-constructive high-rate system energy modeling for battery-powered mobile systems. In: MobiSys'11 the 9th International Conference on Mobile systems, Applications, and Services (2011)
9. Feeney, L., Nilsson, M.: Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In: INFOCOM'01. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (2001)
10. Gartner Research: Gartner Reveals Top Predictions for IT Organizations and Users for 2012 and Beyond (2011). http://www.gartner.com/it/page.jsp?id=1862714
11. Huang, C.M., Hsu, T.H., Hsu, M.F.: Network-aware P2P file sharing over wireless mobile networks. IEEE J. Sel. Areas Commun. **25**, 204–210 (2007)
12. Key Lime 314 LLC: KL Dartboard (2011). https://itunes.apple.com/gb/app/kl-dartboard/id376234917?mt=8
13. Lee, U., Park, J.S., Yeh, J., Pau, G., Gerla, M.: CodeTorrent: content distribution using network coding in VANET. In: MobiShare'06 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking (2006)
14. Mnaoue, A., Shekhar, A.: A generic framework for rapid application development of mobile web services with dynamic workflow management. In: SCC'04 IEEE International Conference on Services Computing (2004)
15. Pajunen, L., Chande, S.: Developing workflow engine for mobile devices. In: EDOC'07 11th IEEE International Enterprise Distributed Object Computing Conference (2007)
16. Pathak, A., Hu, Y.C., Zhang, M.: Where is the energy spent inside my app? fine grained energy accounting on smartphones with eprof. In: EuroSys'12 7th ACM European Conference on Computer Systems. ACM Press (2012)

17. Pentikousis, K.: In search of energy-efficient mobile networking. IEEE Commun. Mag. **48**(1), 95–103 (2010)
18. Philips, E., Carreton, A.L., Joncheere, N., De Meuter, W., Jonckers, V.: Orchestrating nomadic mashups using workflows. In: Mashups '09/'10 the 3rd and 4th International Workshop on Web APIs and Services Mashups (2010)
19. Shachnai, H., Tamir, T.: On two class-constrained versions of the multiple knapsack problem. Algorithmica **29**(3), 442–467 (2001)
20. Sharkey, J.: Coding for life - Battery Life, that is. Google IO Conference (2009)
21. Tang, C., Steinder, M., Spreitzer, M., Pacifici, G.: A scalable application placement controller for enterprise data centers. In: WWW'07 the 16th International Conference on World Wide Web (2007)
22. Zong, Z., Nijim, M., Manzanares, A., Qin, X.: Energy efficient scheduling for parallel applications on mobile clusters. Cluster Comput. **11**(1), 91–113 (2007)